SMC 2016

13th Sound & Music Computing Conference

HAMBURG/GERMANY
31.8.–3.9.2016

CONFERENCE

PROCEEDINGS

S.T.R.E.A.M. FESTIVAL

SMC
Sound & Music Computing
NETWORK

SMC2016.net

Großmann, Rolf | Hajdu, Georg (ed.)

Proceedings SMC 2016 | 31.8.2016, Hamburg, Germany

Published by:
Zentrum für Mikrotonale Musik und Multimediale Komposition (ZM4)
Hochschule für Musik und Theater
http://smc2016.net
Email: chair@smc2016.net

ISSN 2518-3672
ISBN 978-3-00-053700-4

Credits:
Cover design: Veronika Grigkar
Editors: Clarissa Wirth, Madita Wittkopf, Benedict Carey

WELCOME FROM THE CHAIRMAN OF THE PROGRAM COMMITTEE

On behalf of the SMC2016 Program Committee I would like to warmly welcome you to Hamburg and the 13th Sound and Music Computing Conference and Summer School as well as the S.T.R.E.A.M. festival. The summer school, conference and festival are organized by the Hamburg University of Music and Theatre in association with the Hamburg University of Applied Sciences, the University of Hamburg and the Leuphana University Lüneburg.

At this point I would like to express my gratitude to all those who have contributed to this conference: The SMC 2016 Steering Committee and the countless metareviewers, reviewers and subreviewers for their time, advice and opinion that formed a balanced and interesting program, the three co-chairs Rolf Grossmann, Sascha Lemke and Robert Mores for their generous and relentless support, the speakers for preparing presentations that will most certainly inspire us, the composers and sonic artists for their exciting contributions, the editorial team for working hard to enable us to present the program and proceedings before the start of the conference, the local organization committee for their enthusiastic efforts, the VAMH, DEGEM, Kampnagel and Finkenau teams that supported us in various ways and, last but by no means least, our sponsors that have helped us financially to turn this conference into reality.

Georg Hajdu

CHAIRS

Paper Chair
Rolf Grossmann,

Music Chair
Sascha Lemke

Chair of Installations
Robert Mores

Conference Chair
Georg Hajdu

ADVISORY BOARD

Hans-Joachim Braun
Wolfgang Fohl
Rolf Grossmann
Robert Mores
Clemens Wöllner

SMC BOARD

President
Stefania Serafin

Conference Coordinator
Federico Avanzini

Summer School Coordinator
Emilia Gomez

Communication Coordinator
Cumhur Erkut

Web Coordinator
Arshia Cont

Music Coordinator
Juraj Kojs

SMC STEERING COMMITTEE

France
Myriam Desainte-Catherine
Dominique Fober
Gérard Assayag
Yann Orlarey

Italy
Davide Rocchesso
Ricardo Dapello
Federico Avanzini
Pietro Polotti

Greece
Anastasia Georgaki
Ioannis Zannos

Germany
Michael Harenberg
Martin Supper
Stefan Weinzierl
Martin Schüttler

Spain
Xavier Serra
Emilia Gomez

Portugal
Fabien Gouyon
Carlos Guedes
Alvaro Barbosa

Denmark
Stefania Serafin
Jan Larsen

Sweden
Roberto Bresin

United Kingdom
Simon Dixon

ORGANIZING COMMITTEE

Benedict Carey
Jelena Dabic
Daniel Dominguez
Xiao Fu
Philipp Keßling
Goran Lazarevic
Philipp Olbrich
Carlos Rico
Aigerim Seilova
Jacob Sello
Madita Wittkopf
Clarissa Wirth

SUBREVIEWERS, REVIEWERS & METAREVIEWERS

Alessandro Anatrini
Torsten Anders
Luis Antunes
Andreas Arzt
Anders Askenfelt
Federico Avanzini
Stefano Baldan
Jose R Beltran
Emmanouil Benetos
Sebastian Böck
Mattia Bonafini
Sofia Borges
Till Bovermann
Hans-Joachim Braun
Bill Brunson
Ivica Bukvic
Edmund Campion
Sergio Canazza
Yinan Cao
Benedict Carey
Peter Castine
Chris Chafe
Qiangbin Chen
Eric Chou
Se-Lien Chuang
Marko Ciciliani
Fionnuala Conway

John Dack
Alberto de Campo
Stefano Delle Monache
Anthony De Ritis
Myriam Desainte-Catherine
Simon Dixon
Daniel Dominguez
Tony Doyle
Shlomo Dubnov
Bernd Enders
Helmut W. Erdmann
Cumhur Erkut
Bjoern Erlach
Mikael Fernstrom
Arthur Flexer
Federico Fontana
Martin von Frantzius
Jason Freeman
Mike Frengel
Anders Friberg
Henrik Frisk
Peter Gahn
Emilio Gallego
Anastasia Georgaki
Michele Geronazzo
Bruno Giordano
Volker Gnann
Werner Goebl
Masataka Goto
Francesco Grani
Thomas Grill
Rolf Grossmann
Florian Grote
Yupeng Gu
Carlos Guedes
Kerry Hagan
Pierre Hanna
Kjetil Falkenberg Hansen
Sarah-Indriyati Hardjowirogo
Michael Harenberg
Todd Harrop
Mitsuyo Hashida
Folkmar Hein
Joachim Heintz
Hannes Hoelzl
Jan Jacob Hofmann
Risto Holopainen
Andrew Horner
Daniel Hug
Song Hui
Leopold Hurt
Ozgur Izmirli
Dariusz Jackowski
Pierre Jouvelot
Yoshihiro Kanno
Haruhiro Katayose
Damián Keller
Howie Kenty
David Kim-Boyle
Katharina Klement
Volkmar Klien
Peter Knees
Juraj Kojs
Panayiotis Kokoras
Reinhard Kopiez
Andrej Koroliov
George Kosteletos
Johannes S. Kreidler
Johannes Kretz
Mauro Lanza
Goran Lazarevic
Matthias Leimeister
Sascha Lemke
Marcia Lemke-Kern
Stéphane Letz
Hans-Gunter Lock
Lin-Ni Liao
Tapio Lokki
Filipe Cunha Monteiro Lopes
Hanna Lukashevich
Sylvain Marchand
Matija Marolt
Davide Andrea Mauro
Tom Mays
Patrick McGlynn
Annamaria Mesaros
Romain Michon
Julia Mihaly
Kostas Moschos
Dafna Naphtali
Per Anders Nilsson
Vesa Norilo
Ivana Ognjanovic
Konstantina Orlandatou
Daniel Overholt
Rui Pedro Paiva
Stefano Papetti
Richard Parncutt
Dale Parson
Jesper Pedersen
Jussi Pekonen
Luís Antunes Pena
Malte Pelleter
Alfonso Perez
Nils Peters
Jean-Francois Petiot
Marcelo Pimenta
Mark Plumbley
Pietro Polotti
Pedro J. Ponce De León
Laurent Pottier
Carlos Pérez-Sancho
Yagode Quay
Marcelo Queiroz
Rudolf Rabenstein
Laurie Radford
Christopher Raphael
Josh Reiss
Bernard Richardson
Carlos Andres Rico
Jane Rigler
Michal Rinott
Curtis Roads
Antonio Roda'
Francisco Rodríguez Algarra
Julian Rohrhuber
Gerard Roma
Charalampos Saitis
Chris Salter
Augusto Sarti
Greg Schiemer
Andi Schoon
Alexander Schubert
Véronique Sebastien
Sertan Şentürk
Stefania Serafin
Jonathan Shapiro
Matthew Shlomowitz
Johannes S. Sistermanns
Julius Smith
Ryan Ross Smith
Tamara Smyth
Jeffrey Snyder

Jorge Solis
Simone Spagnol
Georgia Spiropoulos
Manfred Stahnke
Ipke Starke
Nikos Stavropoulos
Martin Supper
Gregory Surges
Kenji Suzuki
Pierre-Andre Taillard
Tapio Takala
Akira Takaoka
Hans Tammen
Luis Teixeira
Mari Tervaniemi
Felix Thiesen
Etienne Thoret
Renee Timmers
Giuseppe Torre
Jim Torresen
Yu-Chung Tseng
Shawn Trail
Caroline Traube
Ken Ueno
Marti Umbert
Vesa Valimaki
Leon Van Noorden
Douglas Van Nort
Akito van Troyer
Domenico Vicinanza
Lindsay Vickery
Gualtiero Volpe
Andreas Weixler
Jeremy Wells
Caroline Whiting
Kin Hong Wong
Jim Woodhouse
Lonce Wyse
Clemens Wöllner
Jiajun Yang
Woon Seung Yeo
Steven Yi
Kazuyoshi Yoshii
Jaeseong You
Ioannis Zannos
Massimiliano Zanoni
Ivan Zavada
Paolo Zavagna
Fengyun Zhu

SPONSORS

Hamburg | Behörde für Wissenschaft, Forschung und Gleichstellung
Hamburg | Kulturbehörde
hamburgische kulturstiftung
hochschule für musik und theater hamburg
Co-funded by the Creative Europe Programme of the European Union
OPUS XXI
ULYSSES network
Stanford University
SAIC School of the Art Institute of Chicago
RUDOLF AUGSTEIN STIFTUNG
COMSPOT
YAMAHA
steinberg
KS digital

This conference is supported by the Landesfoschungsförderung Hamburg

HAMBURG SHOWCASE

RESEARCH IN SOUND ANALYSIS AND DESIGN

Wolfgang Fohl & Robert Mores

At the University of Applied Sciences, Hamburg, in the faculty of 'Design, Media and Information' and the faculty of 'Engineering and Computer Science' the areas of research in sound analysis and design are:
Human-computer interaction in virtual acoustic environments – redirected walking, gesture control of virtual sound sources, WFS rendering for interactive environments.
Spatial audio rendering – virtual room acoustics based on measured or computed room impulse responses, WFS enhancements for elevation rendering.
Multimedia – prototypes for 3D-audio-video production workflow.
Sound analysis – instrument identification from guitar sounds. (Wolfgang.Fohl@haw-hamburg.de)
Immersive Audio – 3D sound installations, 3D Audio formats and perception, film sound, audio design and production. (Thomas.Goerne@haw-hamburg.de)
Stringed sound – violin auralization from physical models, high-level violin auralization from acoustical fingerprints taken from Italian masterpieces, pre-manufacturing acoustic design tools for luthiers, timbre perception and representation. (Robert.Mores@haw-hamburg.de)
Interactive Musical Sequences – between notation and compositional thought is as much a gap as between digital music production tools and computer music languages. The Interactive Musical Sequencer combines hierarchic structuring with parametric modification and is capable of generating and reproducing musical content with a familiar and simple user experience and interface. (Philipp.Kessling@haw-hamburg.de)

RECENT RESEARCH PROJECTS CARRIED OUT AT THE INSTITUTE OF SYSTEMATIC MUSICOLOGY

Clemens Wöllner & Tim Ziemer

The Institute of Systematic Musicology at the University of Hamburg is among the largest research centres of its kind in Germany and hosts about 150 students enrolled in BA, MA and PhD programmes. Faculty members have specialized in musical acoustics and music psychology, and taught courses also include popular music studies, empirical aesthetics, audiovisual media and music business as well as sociological and ethnographic approaches to music.

This presentation will briefly focus on recent research projects carried out at the institute. In music psychology, these topics include research on synchronisation, motion capture of musical gestures, human movement sonification, prototypical perception, audiovisual quality judgments, and the study of musical joint actions. In musical acoustics, research covers fields such as sound field synthesis, physical modeling, radiation characteristics of musical instruments, and spatial sound localizations.

RESEARCH, DEVELOPMENT AND PRACTICE OF MUSIC SOFTWARE DESIGN

Georg Hajdu & Panos Kolias

The Hamburg University of Music and Theater (HfMT) has a long tradition in electronic and computer music going back to the mid-1980s. Ever since the establishment of a program in multimedia composition the school became a playground for experimental composers pursuing artistic research projects as well as hardware and software development in areas such as non-standard music notation, new instrument design, networked multimedia performance, sound spatialization as well as interactive music theater.

The growing interest in digital media amongst students in the performing arts programs has not only fostered the teaching of commercial software packages such as Logic or Sibelius, but also increased the need of actively participating in its improvement. The software Melodyne represents such a case allowing both artistic sound manipulation and surgical correction:

Celemony's Melodyne analyzes audio files and allows the user to manipulate single notes (even in harmonic context) in order to create new sounds completely different from the original or at the other extreme, to correct technical issues without affecting the quality or the performance of the original recording.

RESEARCH AND TEACHING IN THE FIELD OF MUSIC AND AUDITORY CULTURE FROM A CULTURAL SCIENCES PERSPECTIVE

Rolf Großmann & Sarah-Indriyati Hardjowirogo

Aesthetic Strategies is an interdisciplinary division within the Institute of Culture and Aesthetics of Digital Media (ICAM) at Leuphana University Lüneburg. ((audio)) is involved in research and teaching (BA and MA Cultural Sciences) in the field of music and auditory culture from a cultural sciences perspective. Research at ((audio)) focuses on questions concerning music and digital media in the sub-fields of (1) technoculture, (2) media integrati-

on, interfaces, surfaces, (3) sampling and program control, and (4) data networks as cultural spaces, thereby contributing to a better understanding of the aesthetics, methods and techniques of digital music production and engaging in theories of media-cultural change. In teaching, ((audio)) puts particular importance on the combination of theoretical education and musical practice. With the Workroom Digital Audio and the audioLab as a high-end production facility, students as well as composers and researchers are provided access to the technological resources required for professional digital audio production. Both research and teaching at ((audio)) reflect the belief that academic education should not only be based on a fixed canon of skills and knowledge but should rather be understood as an organized communication process reflecting both local competence and innovative ability.

The presentation introduces the division's current activities in research and teaching and is complemented by a practical demonstration of recent student works.

MUSICAL AESTHETICS, CREATIVITY, AIRCRAFT NOISE CONTROL

Hans-Joachim Braun

In research on different aspects of sound and music, the Helmut Schmidt University offers a mixed fare. For some time, psychologist Thomas Jacobsen and his group have been researching on experimental music aesthetics, for example on the neural dissociation between musical emotions and liking in experts and laypersons.

A book on creativity in technology and music, edited by Hans-Joachim Braun, has just come out, assessing, inter alia, what cognitive science has to say on creative processes in invention and engineering design and on musical composition and improvisation. What do they have in common?

Andreas Möllenkamp explores the history of music software development and its implications for musical practice and artistic strategies while, in the field of acoustics, Udo Zölzer and his group work on audio coding with short and, hopefully, no delay; on "upmix" from stereo to multi-channel, and, regarding guitar effects, on digital simulation of analog electronic circuits. With a A400M turboprop transport aircraft on campus, Delf Sachau and team have successfully applied an active noise reduction system.

PAPERS AND POSTERS

A STUDY OF THE USE OF PERCEPTUAL FEATURES FOR MUSIC EMOTION RECOGNITION

Maria Abela Scicluna **Adrian Muscat** **Victor Buttigieg**
Department of Computer and Communications Engineering,
Faculty of ICT,
University of Malta, Malta
{maria.abela-scicluna,adrian.muscat,victor.buttigieg}@um.edu.mt

ABSTRACT

Perceptual features are defined as musical descriptors that closely match a listener's understanding of musical characteristics. This paper tackles Music Emotion Recognition through the consideration of three kinds of perceptual feature sets, human rated, computational and modelled features. The human rated features are extracted through a survey and the computational features are estimated directly from the audio signal. Regressive modelling is used to predict the human rated features from the computational features. The latter predicted set constitute the modelled features. The regressive models performed well for all features except for Harmony, Timbre and Melody. All three feature sets are used to train three regression models (one for each set) to predict the components Energy and Valence, which are then used to recognise emotion. The model trained on the rated features performed well for both components. This therefore shows that emotion can be predicted from perceptual features. The models trained on the computational and modelled features performed well in predicting Energy, but not so well in predicting Valence. This is not surprising since the main predictors for Valence are Melody, Harmony and Timbre, which therefore need added or modified computational features that better match human perception.

1. INTRODUCTION

Music feature extraction has traditionally been tackled top-down, using textual, descriptive metadata; or bottom-up, using low-level audio signal descriptors, from which musical concepts may be derived. Unclear or misleading connections between low-level descriptors of the acoustical data, higher level descriptors of the associated musical features, and textual metadata are converging to a boundary that has not yet been overcome. This is termed the *semantic gap* [1].

Studies have shown that several of the shortcomings of the purely data driven techniques can be overcome by applying musical knowledge [1]. It has been proposed that cultural and editorial metadata, currently used by consumers for finding the music they like, be replaced by semantic descriptors within musical dimensions such as rhythm and timbre [2]. The use of meaningful descriptors pushes the *glass ceiling* for music classification to levels higher than originally anticipated for previous data-driven approaches.

This paper undertakes the task of Music Emotion Recognition (MER) through Computational, Rated and Modelled perceptual features. In this paper, the term *feature* is used to denote a particular characteristic of a musical extract. Features were either estimated directly from the audio signal by signal-processing tools, rated perceptually by listeners, or modelled using pattern recognition algorithms. To distinguish between the three groups, they will be called Computational, Rated and Modelled features respectively, to emphasise their respective derivation approach. A dataset of musical extracts was compiled for this purpose.

This study goes a step beyond the current trend in MER of classifying emotion directly from the available computational features or social tags. Rated features are modelled from Computational features and then used in the MER task. The reason behind this step is to reduce the large number of proposed features in the literature to a smaller list of features that can be distinguished by music listeners, irrespective of their musical background. The performance of the different feature sets is compared through the task of music emotion classification.

Section 2 gives an overview on related work in the area. Section 3 describes the framework built for this study and gives an overview of each processing block. Observations and results of every processing block are given in Section 4. A brief discussion follows in Section 5 and conclusions are drawn in Section 6.

2. RELATED WORK

Psychological studies have shown that emotions conveyed by music are objective enough to be valid for mathematical modelling [3]. The main efforts of emotion-based music retrieval have focused on discovering signal features that are correlated with the affective scores. In order to classify music emotion, preference is given to selecting those features directly related to the human perception of music [4].

Cook [5] states that the musical aspects that humans use to describe music are pitch, loudness, duration, timbre, style and texture. However, the information directly repre-

sented in the acoustic signal is only the physical property of music, such as absolute pitch and note duration. Herrera et al. [2] propose the incorporation of higher-level semantic descriptors to a given feature set. They describe semantic descriptors as measures that can be computed directly from the audio signal, by means of the combination of signal processing, machine learning techniques, and musical knowledge. Their goal is to emphasise the musical attributes of audio signals (such as rhythm and instrumentation), attaining higher levels of semantic complexity than low-level features (such as spectral centroid and spectral flux), but without being restricted by the rules of music notation, since human music perception retrieves information that might be different from traditional concepts of music theory.

The approach taken by Hedblad [6] for the modelling of perceptual music descriptors, was to include all extracted features as potential predictors and apply the stepwise version of linear regression. However, non-linear relationships were not taken into account during the modelling process. In other areas of Music Information Retrieval (MIR), Support Vector Regression (SVR) and Classification (SVC) is a widely used supervised learning classification algorithm. It has been found superior to other machine learning methods in a number of studies [7–9].

3. METHODOLOGY

The framework built for this study is shown in Figure 1 depicting each processing block as well as input and output data as required for each phase. Each processing block is further explained in the following sub-sections.

3.1 Dataset Selection

Figure 2 displays the Energy-Valence (EV) Model for Emotion which was built for this study. It is based on Thayer's two-dimensional emotion model [10], with descriptors from Russell's circumplex model of affect [11] and their adaptations, which were used in [3, 4, 9, 12, 13]. An emotive descriptor was chosen for the horizontal and vertical axes and each diagonal. The chosen descriptors were simple and distinct adjectives to avoid ambiguity. Using this model, the dataset was chosen such that the selected music clips would be uniformly distributed across the Energy-Valence plane.

Four online music databases which offer a search by mood (All Music Guide [14], Last.fm [15], Spotify [16] and Stereomood [17]) were identified. These were also used in other published work related to MER [3, 9, 13]. The first five songs suggested by each database for every emotion were selected. A thirty-second segment was then manually extracted from the refrain. During this process, the music clips were converted to a uniform format (44,100 Hz, 16 bits, stereo PCM WAV) and normalised to the same volume level. Clips were amplified such that the peak amplitude was 0dB. Two-second fade-in and fade-outs were introduced to improve the listening experience. The music clips were kept in their original form, irrespective of whether they were instrumental or included lyrics, since changing instrumentation was considered as changing the listening experience, a fact that might have a negative effect on the modelling and classification phases. The dataset was then fine tuned to ensure emotion consistency in the music clips, discarding excerpts with explicit and dominant positive or negative lyrics that would influence the ratings and limiting overly familiar songs. The final dataset was composed of seventy, thirty-second music extracts.

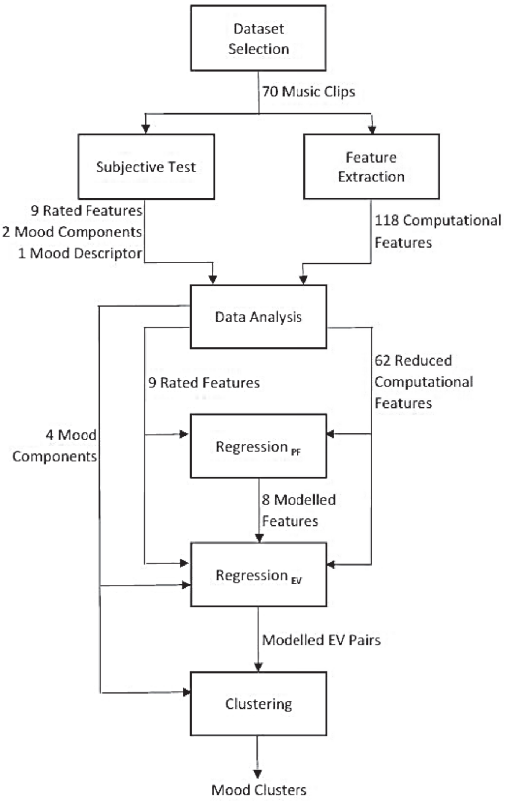


Figure 1: Schematic diagram of the framework built for the required tests and analysis, depicting each processing block as well as input and output data as required for each phase. Two instances of regression were performed. The first ($Regression_{PF}$), describes the modelling of Rated features through the Computational ones. The second regression process ($Regression_{EV}$), describes modelling of the Energy and Valence emotive components through three separate feature sets.

3.2 The Subjective Test

A subjective test was conducted through a computer-based survey. It was distributed by email to volunteers who showed interest in completing it. While they were instructed to complete it in a quiet environment it was not run in a controlled environment, in order to approximate the participant's preferred music-listening habits. Each survey participant was asked to listen to thirty-five music clips played in random order. Each participant had to grade each music clip using eleven music features. This was done through a slider with extremes at each end, as depicted in Figure 3. To calculate the mean rating, the slider position was later translated to an integer between

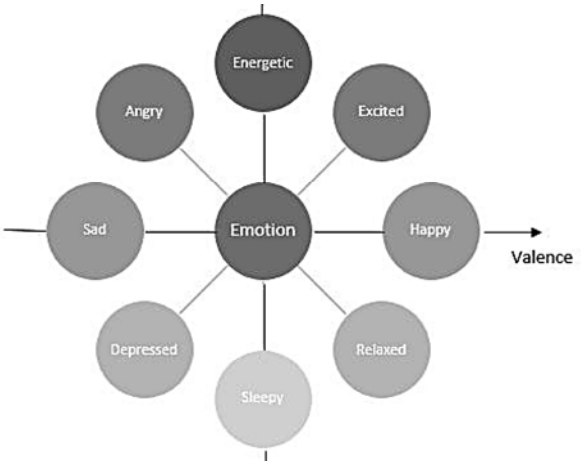


Figure 2: The Energy-Valence Model for Emotion that was compiled for this study. Simple and distinct emotive descriptors were selected.

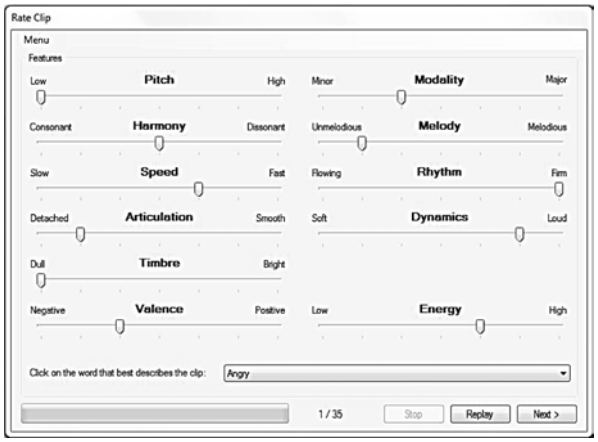


Figure 3: The graphical user interface used for rating the music clips in the survey.

1 and 7, with 4 being regarded as the neutral rating. The test structure was based on similar rating tests for perceptual features [18, 19]. Features in these tests were in turn inspired by previous published work [20]. The Rated music features were nine perceptual features (Pitch, Modality, Harmony, Melody, Speed, Rhythm, Articulation, Dynamics and Timbre) and two emotive descriptors ($\text{Energy}_{\text{Rated}}$ and $\text{Valence}_{\text{Rated}}$), as detailed in [21]. A brief explanation and ten-second samples were given to the participants in an introductory screen.

Each music clip was also labelled using one of the eight emotions given in Figure 2. This was achieved using a drop-down menu. The aim of these two approaches was to confirm that consistent emotive results would be obtained. In order to calculate the average response of the emotive descriptor, each descriptor was translated to a pair of co-ordinates such as to represent their circular distribution as shown in Figure 2. The descriptor was split in two components representing the Energy and Valence dimensions. These components will further be referenced as $\text{Energy}_{\text{Mood}}$ and $\text{Valence}_{\text{Mood}}$ respectively. This split facilitates the comparison with the linear ratings, further referenced as $\text{Energy}_{\text{Rated}}$ and $\text{Valence}_{\text{Rated}}$.

The seventy music extracts were split in two groups of thirty-five clips, both uniformly distributed along the EV model. Thirty participants rated one of these two groups. Another five participants voluntarily rated the whole dataset, rating each group on different days. Eighteen participants were males and seventeen were females and they were aged between 22 and 63 (average age of 35). Twelve had never practised or studied music, eight had basic or informal music training and fifteen had advanced music background.

3.3 Computational Feature Extraction

Four tools (Essentia [22], MIRtoolbox [23], PsySound3 [24] and Sonic Annotator [25]) were chosen to extract 118 Computational features that may be used to approximate perceptual features, as detailed in [21]. The scope of this study is not to judge the accuracy of a particular feature extraction algorithm; but rather, how close it comes to a perceptual descriptor. Feature extraction was done through a short-term window (frame) that moves chronologically along the temporal signal. The frame length varies across algorithms, but is generally in the order of tens of milliseconds with hop size (overlap) of half its length. The results were summarised by taking means and variances across the frames.

3.4 Data Analysis

Data Analysis was performed through IBM SPSS Statistics 22 [26]. For the survey results, values for Cronbach's alpha, mean inter-item correlation, histograms for mean ratings, Friedman test and error bar graphs were extracted. Dimension reduction through Principal Component Analysis (PCA) was performed after the computational feature extraction to avoid redundancy of multiple features measuring the same thing and for summarising a number of linearly highly correlating features. This reduced the number of Computational features to 62.

3.5 Feature Modelling

Models for each of the Rated features were built from the Computational features through Multiple Linear Regression (MLR), Artificial Neural Networks (ANN) and Support Vector Regression (SVR), as depicted by $Regression_{PF}$ in Figure 1. Feature selection is incorporated in linear modelling through the Weka M5 option. For ANN and SVR, the feature selection algorithm ReliefF [27] was used to identify the most relevant Computational features for each of the predicted variables. The model parameters were selected through ten iterations of 10-fold cross-validation using Weka Experimenter [28]. The main statistical value that was minimised during the parameter estimation process is the root mean squared error (RMSE). Once the RMSE for a particular model was determined, the correlation coefficient r was noted. The models were compared based on these two attributes. The

Weka ZeroR classifier [28] was used to extract the mean model, taken in this study as the Baseline model.

3.6 Emotion Classification

Emotion classification was utilised as a way to compare performance of the various types of perceptual feature sets. This was performed in two steps. The first step, depicted by $Regression_{EV}$ in Figure 1, was to model the Energy and Valence components through SVR. Regression was attempted with three different feature sets: Rated (the nine features rated by the survey participants, as described in Section 3.2), Modelled (the eight features modelled through SVR, as described in Section 3.5) and Computational (sixty-two Computational features obtained after the Computational feature extraction and data reduction as described in Section 3.3).

The second step was to apply the k-means clustering algorithm on each resulting Energy-Valence model. Four clusters were extracted each time. The choice of four clusters even though there were originally eight emotive descriptors allows for some tolerance to the subjectivity of emotions.

The average song ratings for the emotion components $\text{Energy}_{\text{Rated}}$, $\text{Valence}_{\text{Rated}}$ and $\text{Energy}_{\text{Mood}}$, $\text{Valence}_{\text{Mood}}$ were used as inputs for the k-means clustering algorithm. The outputs were taken as the ground truth cluster ownership for each song. A classification was considered to be accurate when a song was clustered within the same group as the respective ground truth. Performance was compared through scatter plot distribution, cluster centroid centres, and clips included in each cluster.

4. RESULTS

4.1 The Subjective Test

Values for Cronbach's alpha varied between 0.84 (for Modality) and 0.98 (for Speed), confirming internal consistency of the survey. The p-value for the Friedman Tests for every feature was in the order between 10^{-37} and 10^{-160}, which means that all mean feature rating scores differ significantly between song extracts. However this does not imply that all of the seventy songs differ significantly from each other in every feature, but that groups (or clusters) can be identified.

Inter-rater correlation was not consistent for all features but differed according to the type of feature and difficulty of the task. Observations indicate that the more difficult the feature meaning was to grasp, the closer to the neutral rating (4) the majority of the ratings for the feature were.

The distribution of mean ratings splits the features in three groups with similar observed characteristics. Participants found features in the first group (Speed, Dynamics, Energy) to be the easiest to rate and agreed between each other to a large extent, with a mean inter-rater correlation of 0.72, 0.65 and 0.66 respectively. Values for mean ratings in this group were the furthest from the neutral rating (4), implying high agreement between raters. These three features are common adjectives for music description. They are also the least subjective, since stating a song as being fast is independent of whether one prefers fast or slow songs.

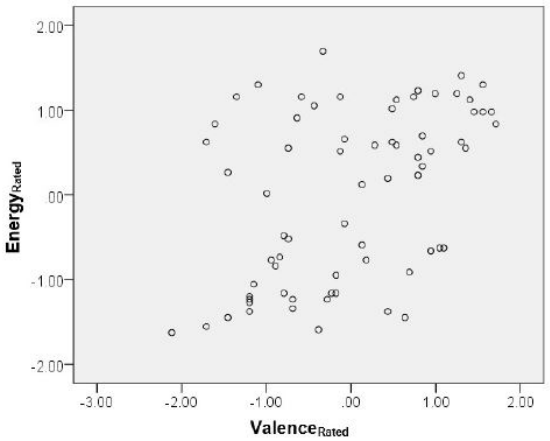
The values for the inter-rater correlation for features in the second group (Valence, Timbre, Pitch, Melody, Harmony), were between 0.29 and 0.40. Though less common, and more subjective, participants seemed to understand the meaning of these five descriptors. Features in the third group (Articulation, Rhythm, Modality) were the hardest for listeners to rate. Their descriptions are more technical and they are the least natural for people to notice or comment about, unless one is specifically listening to that particular aspect. The observed distributions, where averages are located toward the middle of the range, is indicative of a lack of agreement between listeners. Weak inter-rater correlation was observed with values of 0.25, 0.20 and 0.26 respectively. However, though in this case these are a small minority, extremes can still be distinguished.

The results for the mean inter-rater correlations were compared with those published for a similar survey in [18], where participants all had a level of musical background. The values show that participants with considerable musical experience improve results for the more technical features (Rhythm, Modality and Articulation) but not for the others. On the other hand, keeping the original instrumentation of the music extracts improved the results for features where instrumentation is essential (Timbre, Dynamics and hence, Energy).

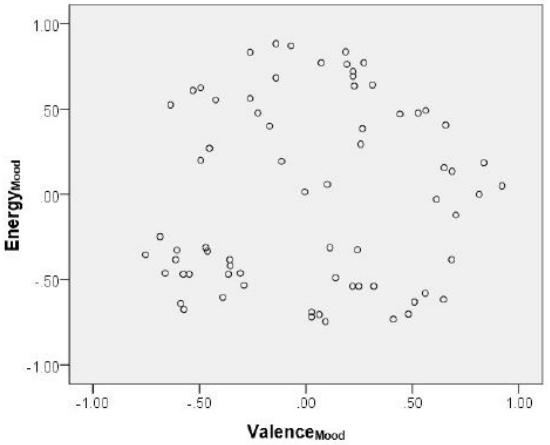
Linearity was confirmed between the components of the emotive descriptors ($\text{Energy}_{\text{Mood}}$ and $\text{Valence}_{\text{Mood}}$), and $\text{Energy}_{\text{Rated}}$ and $\text{Valence}_{\text{Rated}}$ respectively, with correlation values between these two pairs of attributes exceeding 0.9. The scatter plots in Figures 4 (a) and (b) illustrate that the distribution for the emotive descriptors (Figure 4 (b)) is closer to the desired circular shape. Clusters per quadrant seem identifiable, with few music clips being located towards the centre of the distribution. This indicates that there were only few cases where ratings were so varied that the calculated average was close to 0. This is a further indication of rater agreement. The scatter plot for $\text{Energy}_{\text{Rated}}$ and $\text{Valence}_{\text{Rated}}$ (Figure 4 (a)) hints at a skew in the positive diagonal, with values expected in the calm-positive quadrant overlapping the calm-negative quadrant.

4.2 Feature Analysis

Significant correlations were identified between algorithms extracting similar Computational features (refer to Section 3.3). Lack of perfect correlation was attributed to different approaches and different parameters. However, when these were compared to the Rated features, correlation was weaker. The highest correlation values between Rated and Computational features values were obtained for Dynamics (0.80), Speed (0.74) and Pitch (0.56). Articulation did not correlate significantly with any of the respective Computational features. The highest correlation between the Rated features Timbre, Rhythm and Harmony and the corresponding Computational features was between 0.31 and 0.46. This explains why other studies [13, 19] extract features using different algorithms but



(a) A Plot of Rated Energy and Valence



(b) A Plot of the Emotive Descriptors

Figure 4: Scatter plots displaying the distribution of the Rated emotion components $\text{Energy}_{\text{Rated}}$ and $\text{Valence}_{\text{Rated}}$ and the components of the emotive descriptors, $\text{Energy}_{\text{Mood}}$ and $\text{Valence}_{\text{Mood}}$.

return similar classification results.

4.3 Feature Modelling

As was also observed by Yang et al. [8], non-linear regression proved to be superior to the linear alternative to model human rated perceptual features. Table 1 displays the returned correlation coefficient values for the three regression algorithms. The models fared worst when only five features were included, while the optimal number of features was found to be between ten and twenty-five. In almost all cases, results for the non-linear models, that is, ANN and SVR, were very similar, with SVR faring slightly better. The worse performance of MLR may be attributed to leaving only the linearly correlating features for the model, so any non-linear relationship was not taken into account in this model. In all cases the best model was found to be through SVR.

Acceptable models were built through SVR for the majority of perceptual descriptors, as shown in Tables 1 and 2. The features with best inter-rater agreement, Dynamics and Speed, were modelled best, with correlation coefficients between observed and modelled values of 0.93 and 0.89 respectively. However, the RMSE values obtained for Timbre and Harmony (0.8) were hardly better than the baseline (1). Since these were not the features with worst inter-rater agreement, the reason for the bad results was attributed to a general mismatch between Computational and Rated features.

Modelled Feature	Model 1 MLR	Model 2 ANN	Model 3 SVR
Speed	0.82 (0.13)	0.88 (0.08)	0.89 (0.08)
Rhythm	0.74 (0.21)	0.77 (0.17)	0.80 (0.15)
Pitch	0.64 (0.22)	0.72 (0.21)	0.72 (0.22)
Modality	0.57 (0.25)	0.65 (0.21)	0.78 (0.15)
Harmony	0.48 (0.31)	0.47 (0.31)	0.54 (0.28)
Melody	No model better than baseline found.		
Dynamics	0.87 (0.09)	0.92 (0.06)	0.93 (0.06)
Articulation	0.67 (0.23)	0.67 (0.24)	0.69 (0.22)
Timbre	No model better than baseline found.		0.63 (0.18)

Table 1: Correlation coefficient (r) values for feature modelling using the three algorithms: MLR, ANN and SVR. No one algorithm stands out as significantly better than the rest though best values are obtained through SVR. The standard deviation is given in brackets.

Modelled Feature	Number of Features	RMSE
Speed	20	0.45 (0.10)
Rhythm	15	0.56 (0.18)
Pitch	15	0.68 (0.19)
Mode	20	0.61 (0.14)
Harmony	20	0.82 (0.16)
Melody	No model better than baseline found.	
Dynamics	25	0.37 (0.10)
Articulation	10	0.74 (0.14)
Timbre	10	0.83 (0.20)

Table 2: The number of features included in the best model and RMSE values for feature modelling through SVR. The standard deviation is given in brackets. The optimal number of computational features included in the model was found by altering the amount after feature selection. The model with the lowest RMSE and lowest number of features was chosen.

Computational features directly related to the Rated ones were selected through feature selection in the respective models in the majority of cases. However, a number of Computational features from other musical domains, particularly the spectral domain, were also selected in the models. These facts, in addition to the weak correlation between the respective Rated and Computational features confirm that though progress is being made in modelling of perceptual musical descriptors, approximating a single computational feature to its perceptual equivalent is still an open research question.

Emotive Component	Rated	Modelled	Computational
$\text{Valence}_{\text{Rated}}$	0.95 (0.05)	0.57 (0.24)	0.68 (0.20)
$\text{Energy}_{\text{Rated}}$	0.98 (0.01)	0.89 (0.08)	0.86 (0.09)
$\text{Valence}_{\text{Mood}}$	0.87 (0.09)	0.53 (0.30)	0.61 (0.26)
$\text{Energy}_{\text{Mood}}$	0.95 (0.04)	0.87 (0.09)	0.85 (0.10)

Table 3: Correlation coefficient (r) values for the modelled components of emotion using three separate feature sets. The Valence component consistently returns weaker results, but are significantly better for the Rated feature set. Values for the standard deviation are included in brackets.

4.4 Emotion Classification

4.4.1 Regression

Table 3 displays the correlation coefficient values for the different feature sets used for modelling the emotion components. Better results were obtained for $\text{Energy}_{\text{Rated}}$ and $\text{Valence}_{\text{Rated}}$ when compared to $\text{Energy}_{\text{Mood}}$ and $\text{Valence}_{\text{Mood}}$ respectively. This can be explained by the similarity in rating method of the former pair to the remaining features.

For all components, the best feature set is the Rated one. Worse results were obtained for the Modelled and Computational feature sets. While results for the Energy components, with correlation coefficient values above 0.85, are acceptable, r values for the Valence components, particularly in the Modelled feature set (less than 0.6) are weak. Moreover, while for the Energy components, the Modelled feature set fared slightly better than the Computational one, for the Valence components, it fared slightly worse.

The difference in performance between the Energy and Valence dimensions is explained by which features affect the respective dimension. The highest ranking features for the Energy components, Speed and Dynamics, were the ones which were best modelled in the previous phase (with r values of 0.89 and 0.93 respectively). Results for the models of the three main predictors for Valence were not as good. While a model for Melody was not built due to high RMSE values, models for Timbre and Harmony, had the lowest r (0.63, 0.54) and highest RMSE (0.83, 0.82) values. Despite this high margin of error, these models were used to compile the Modelled feature set.

Better results obtained for the Energy component means that the ambiguity and difficulty in MER brought about by the subjectivity of emotions is focused in the Valence dimension, since accurate results can be and have been achieved with current algorithms in the Energy dimension. Still, $\text{Valence}_{\text{Rated}}$ reached an r value of 0.95 for the Rated feature set. This means that accurate modelling is possible with the right feature set. However, such values have not been reached using Computational features. The problem therefore lies in identifying the Computational features that can accurately model the perceptual features effecting Valence. Though still complex, modelling a perceptual feature of a song is still less subjective than modelling its positivity.

Cluster	Feature Set		
	Rated	Modelled	Computational
Energetic-Positive	95.8%	70.8%	83.3%
Calm-Positive	80.0%	20.0%	66.7%
Calm-Negative	82.4%	64.7%	52.9%
Energetic-Negative	85.7%	85.7%	85.7%
Average	86.0%	60.3%	72.2%

Table 4: Cluster accuracy for the Rated emotion components $\text{Energy}_{\text{Rated}}$ and $\text{Valence}_{\text{Rated}}$ obtained through the three feature sets.

Cluster	Feature Set		
	Rated	Modelled	Computational
Energetic-Positive	71.4%	76.2%	61.9%
Calm-Positive	64.7%	11.8%	76.5%
Calm-Negative	94.1%	76.5%	64.7%
Energetic-Negative	86.7%	80.0%	66.7%
Average	79.2%	61.1%	67.4%

Table 5: Cluster accuracy for the components of the emotion descriptors $\text{Energy}_{\text{Mood}}$ and $\text{Valence}_{\text{Mood}}$ obtained through the three feature sets.

4.4.2 Clustering

The above observations were confirmed by the clustering results, displayed in Tables 4 and 5. Average accuracy ranges between 60% and 88%. When the Valence dimension was removed, that is, combining negative and positive options and considering only the calm and energetic clusters, accuracy rate exceeded 90% for all feature sets.

As can be observed in Tables 4 and 5, the biggest misclassifications are located in the calm clusters. For the Modelled feature set, where the calm-positive cluster has the smallest accuracy, the music clips were clustered with the calm-negative clips. This shows a bigger disagreement on whether calmer songs are positive (relaxed) or negative (depressed). The results may indicate that this differentiation relies more on taste than the energetic equivalents. People who enjoy songs in the energetic-negative cluster, tend to still describe them as angry songs. However, people who enjoy calmer songs describe them as relaxed but for those who do not, they are depressive.

This further highlights that the subjectivity of emotions focused in the Valence dimension, is particularly present in calmer music. The better results obtained by the Rated feature sets when compared with the Computational and Modelled ones indicate the importance of focusing on the listener for efficient classification systems.

5. DISCUSSION

Though acceptable results were obtained in the subjective test, a more consistent inter-rater correlation across all musical descriptors is desirable to ensure accurate modelling. Since the ground truth is based on these ratings, improvement in model accuracy will be difficult if these are not optimal. The starting point is grasping the human per-

ceptual understanding through optimally constructed subjective tests. The dataset needs to be adapted to the concept being tackled to highlight variances and observe similarities throughout genres. Having a subjective test per modelled feature might also improve inter-rater correlation since participants will focus on identifying a single feature, rather than eleven different elements.

A more in-depth analysis of the computational algorithms needs to be performed, and where required, adjusted or new ones developed. From a perceptual point of view, the computational features need to be a derivative of the perceptual features rather than the other way round. For better perceptual accuracy, the computational features need to go beyond theoretical correctness as in classical music theory and get closer to what people understand by the descriptor. A good example of this is perceptual speed against theoretical tempo. As shown in [29], tempo alone cannot accurately represent perceptual speed.

The work described in this paper shows that perceptual machine learned models can accurately predict human perception of music. Models for Speed and Dynamics made use of features beyond their own musical domain, which rendered the prediction truly representative of the perceptual equivalent. Subsequently, these two models contributed positively to the energy component of emotion, resulting in high accuracy ($r = 0.89$). This means that the model developed in this study accurately estimated human understanding, implying a reduction in the semantic gap.

6. CONCLUSION

The process of breaking down music emotion recognition into subtasks for the identification of the perceptual components led to a clearer understanding of the problem at hand. This study found that the first step required to tackle subjectivity in music emotion is to focus on the perceptual predictors of the Valence component. Though still complex, modelling a perceptual feature should be less subjective than modelling positivity. Similarly, other higher level classification concepts, such as Location or Activity, are bound to human understanding. Hence, the pursuit of accurate modelling of perceptual features is a natural approach.

The analysis in this paper highlights future points of focus required for narrowing the semantic gap. Once accurate computational perceptual features are identified, the large number of currently existing computational features is reduced to a much smaller number, close to human understanding. Rated and computational features such as Speed, Dynamics, Brightness and Dissonance will be highly correlated such that one computational perceptual feature will accurately represent the respective Rated feature. At this stage, addressing classification of higher level concepts can be reduced to the identification of the best out of about ten features, thus heavily simplifying these tasks.

7. REFERENCES

[1] Ò. Celma, P. Herrera, and X. Serra, "Bridging the music semantic gap," 2006.

[2] P. Herrera, J. Bello, G. Widmer, M. Sandler, Ò. Celma, F. Vignoli, E. Pampalk, P. Cano, S. Pauws, and X. Serra, "Simac: Semantic interaction with music audio contents," in *The IET Second European Workshop on the Integration of Knowledge, Semantics and Digital Media Technology, London, UK*, 2005, pp. 399–406.

[3] C. Laurier, O. Meyers, J. Serrà, M. Blech, P. Herrera, and X. Serra, "Indexing music by mood: Design and integration of an automatic content-based annotator," *Multimedia Tools and Applications*, vol. 48, no. 1, pp. 161–184, 2010.

[4] W. Muyuan, Z. Naiyao, and Z. Hancheng, "User-adaptive music emotion recognition," in *IEEE Seventh International Conference on Signal Processing (ICSP), Beijing*, vol. 2, 2004, pp. 1352–1355.

[5] P. R. Cook, *Music, cognition, and computerized sound*. Cambridge, MA: Mit Press, 1999.

[6] A. Hedblad, "Evaluation of musical feature extraction tools using perceptual ratings." Master's thesis, KTH Computer Science and Communications, Stockholm, Sweden, 2011.

[7] N. Wack, E. Guaus, C. Laurier, O. Meyers, R. Marxer, D. Bogdanov, J. Serra, and P. Herrera, "Music type groupers (mtg): generic music classification algorithms," *Music Information Retrieval Evaluation eXchange (MIREX) extended abstract*, 2009.

[8] Y.-H. Yang, Y.-C. Lin, Y.-F. Su, and H. H. Chen, "A regression approach to music emotion recognition," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 16, no. 2, pp. 448–457, 2008.

[9] B.-j. Han, S. Ho, R. B. Dannenberg, and E. Hwang, "Smers: Music emotion recognition using support vector regression," in *Proceedings of the ninth International Society for Music Information Retrieval (ISMIR) Conference*, 2009, pp. 651–656.

[10] R. E. Thayer, *The biopsychology of mood and arousal*. Oxford University Press, 1989.

[11] J. A. Russell, "A circumplex model of affect," *Journal of personality and social psychology*, vol. 39, no. 6, p. 1161, 1980.

[12] J. I. Lee, D.-G. Yeo, B. M. Kim, and H.-Y. Lee, "Automatic music mood detection through musical structure analysis," in *IEEE Second International Conference on Computer Science and its Applications (CSA)*, 2009, pp. 1–6.

[13] A. S. Bhat, V. Amith, N. S. Prasad, and D. M. Mohan, "An efficient classification algorithm for music mood detection in western and hindi music using audio feature extraction," in *IEEE Fifth International Conference on Signal and Image Processing, Karnataka, India*, 2014, pp. 359–364.

[14] "The All Music Guide," http://www.allmusic.com/, accessed: October 2014 to May 2015.

[15] "last.fm," http://www.last.fm, accessed: October 2014 to May 2015.

[16] "Spotify," https://www.spotify.com, accessed: October 2014 to May 2015.

[17] "Stereomood," http://www.stereomood.com/, accessed: October 2014 to May 2015.

[18] A. Friberg, E. Schoonderwaldt, and A. Hedblad, "Perceptual ratings of musical parameters," *Gemessene Interpretation-Computergestützte Aufführungsanalyse im Kreuzverhör der Disziplinen, Mainz: Schott*, pp. 237–253, 2011.

[19] A. Friberg, E. Schoonderwaldt, A. Hedblad, M. Fabiani, and A. Elowsson, "Using listener-based perceptual features as intermediate representations in music information retrieval," *The Journal of the Acoustical Society of America*, vol. 136, no. 4, pp. 1951–1963, 2014.

[20] L. Wedin, "A multidimensional study of perceptual-emotional qualities in music," *Scandinavian journal of psychology*, vol. 13, no. 1, pp. 241–257, 1972.

[21] M. A. Scicluna, "A study of automated feature extraction and classification for music emotion recognition," Master's thesis, University of Malta, 2015.

[22] D. Bogdanov, N. Wack, E. Gómez, S. Gulati, P. Herrera, O. Mayor, G. Roma, J. Salamon, J. R. Zapata, and X. Serra, "Essentia: An audio analysis library for music information retrieval." in *Proceedings of the 14th International Society for Music Information Retrieval (ISMIR) Conference, Brazil*, 2013, pp. 493–498.

[23] O. Lartillot and P. Toiviainen, "A MATLAB toolbox for musical feature extraction from audio," in *International Conference on Digital Audio Effects, Bordeaux, France*, 2007, pp. 237–244.

[24] D. Cabrera *et al.*, "Psysound: A computer program for psychoacoustical analysis," in *Proceedings of the Australian Acoustical Society Conference*, vol. 24, 1999, pp. 47–54.

[25] C. Cannam, M. O. Jewell, C. Rhodes, M. Sandler, and M. d'Inverno, "Linked data and you: Bringing music research software into the semantic web," *Journal of New Music Research*, vol. 39, no. 4, pp. 313–325, 2010.

[26] "SPSS Statistics 22," software, 2013, IBM.

[27] I. Kononenko, E. Šimec, and M. Robnik-Šikonja, "Overcoming the myopia of inductive learning algorithms with relieff," *Applied Intelligence*, vol. 7, no. 1, pp. 39–55, 1997.

[28] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, and I. H. Witten, "The WEKA data mining software: an update," *ACM SIGKDD explorations newsletter*, vol. 11, no. 1, pp. 10–18, 2009.

[29] A. Elowsson, A. Friberg, G. Madison, and J. Paulin, "Modelling the speed of music using features from harmonic/percussive separated audio," in *Proceedings of the 14th International Society for Music Information Retrieval (ISMIR) Conference*, 2013, pp. 481–486.

SONIFICATION AS CATALYST IN TRAINING MANUAL WHEELCHAIR OPERATION FOR SPORTS AND EVERYDAY LIFE

Andreas Almqvist Gref
KTH Royal Institute of Technology
aalmqvi@kth.se

Ludvig Elblaus
KTH Royal Institute of Technology
elblaus@kth.se

Kjetil Falkenberg Hansen
KTH Royal Institute of Technology
Södertörn University
kjetil@kth.se

ABSTRACT

In this paper, a study on sonification of manual wheelchair movements is presented. The aim was to contribute to both rehabilitation contexts and in wheelchair sports contexts, by providing meaningful auditory feedback for training of manual wheelchair operation. A mapping approach was used where key parameters of manual wheelchair maneuvering were directly mapped to different sound models. The system was evaluated with a qualitative approach in experiments. The results indicate that there is promise in utilizing sonification for training of manual wheelchair operation but that the approach of direct sonification, as opposed to sonification of the deviation from a predefined goal, was not fully successful. Participants reported that there was a clear connection between their wheelchair operation and the auditory feedback, which indicates the possibility of using the system in some, but not all, wheelchair training contexts.

1. INTRODUCTION

Our perception of sound is linked to our understanding of physical properties of objects, as well as our understanding of how objects interact and move in the physical world [1, 2]. In music, the gestural performance and its link to sound and bodily movement has been the subject for much research and has been shown effective by Wanderley among others [3,4], and performers' movements are moreover affected by the instant audio feedback from bodily interaction with the instrument in a closed-loop sonification [5].

Studies of sonification of body movements have shown that this type of feedback may improve motor task learning by making movement relations more obvious to the user of the system [6]. This can be done either by giving feedback on the deviation from a desired movement or by a more time demanding approach where the feedback is linked not to the deviation but to the movement in a more direct sense, and thus give guidance towards a goal which is apparent to the user by already present stimuli or by the assistance of a trainer.

In the presented study the possibilities of auditory feedback for the purpose of training manual wheelchair movements are studied. The targeted training context is that of expert supervised exercises in a gym, where precision is valued higher than portability. There are two main fields that may be considered; firstly in everyday-life and rehabilitation contexts relating mostly to learning fundamental movement technique, and secondly, movement in wheelchair sports. Our aims are partly to investigate sonification as catalyst and motivation in repetitive training, and partly to investigate audio feedback in training of accurate movements where other bio-feedback is insufficient.

Manual wheelchairs are used by a variety of persons who are partly or fully dependent on an assistive device for locomotion. The only property these users share are that they are able to use their arms for propelling the wheelchair. While standard wheelchairs may be similar in features, the group of mobile wheelchair riders have a smaller base of common ground than what might be the first assumption. Depending on the disability, operation will be done differently. For example, if the disability inhibits from leaning forward, as with high positioned spinal cord injury, both wheelchair propulsion and the fundamental skill of back wheel balance becomes significantly more difficult [7].

The most basic control of a wheelchair is done through grabbing the push rings of the back wheels and pushing the wheels in synchrony to gain forward propulsion. Applying more force to one wheel than the other allows for turning. A wheelchair for everyday-life use needs to be able to cross obstacles such as door frames and curbs by tilting the chair backwards into a back wheel balancing position. Sport wheelchairs have different design constraints than that of the everyday-life wheelchair. Instead of front wheels, most of these sport chairs have small caster wheels, two in front and two in back. Back wheel balancing is still important however, with all four casters hovering slightly above ground. Maintaining this position eases both turning and propulsion. With good technique this will allow for turning without using hands on the push rings.

In this work the sonification definition proposed by Hermann [8] is used, which entails four major criteria: that the sound produced reflects *objective* properties of the input data, that the transformation to sound is *systematic*, and that this transformation is *reproducible* meaning that the output sound is consistent and structurally identical given the same input data. Additionally the sonification system must be *general* and possible to use for other data.

1.1 Strategies for Feedback

In relation to motor task learning, the sonification definition criteria of reproducible and systematic sonification is especially important. Without a consistent feedback the interaction loses its ability to aid in gaining insight [8–10]. Sonification of bodily movements or human activity in general may, if the above criteria is met, be used to clarify goal attainment without the explicit inclusion of goal specifics into the sound model [8]. The sound then acts as periphery guidance towards a self defined goal or a goal which, the attainment of, is evident by another modality. Hermann argues [11] that sonification of body movements may act as a way for the user of the system to gain insight into the movements performed. The sonification allows for monitoring of ones activities and consequently to evaluate differences in gesture execution.

Previous work show that sonification and auditory feedback improve motor task learning of complex movements [9,10] and that auditory feedback can enhance both perception accuracy and reproduction accuracy of complex sport movements [6]. Furthermore, the use of auditory feedback as a partial replacement for visual and sensory information has been shown effective for maintaining balance in upright stance [12], indicating that auditory bio feedback of deviations from a desired movement may help proprioception. Sigrist argues [10] that feedback for motor task learning should be designed as to direct the learner to the already present information that is most relevant for the execution of the movement. Sonification in therapy can also help to examine movements of a patient [13].

In summary, when learning complex movements or motor tasks, one must receive continuous feedback of the performance. The strategy of sonification becomes important because it must relate to the goal of the system. Sigrist outlines three main approaches [14]: 1) Auditory alarms, meaning discrete feedback when a predefined threshold is exceeded; 2) Sonification of movement variables, where movement continuously controls auditory feedback; and 3) Sonification of movement error, meaning continuous sonification of the deviation from a predefined movement pattern. Auditory alarms are discontinuous and less likely to stimulate complex motor task learning. Feedback on movement variables will depend on the user to have knowledge of a properly performed movement. Sonifying movement error will demand detailed knowledge of the characteristics of a correct movement to be built into the system.

No particular strategy from previous studies was adopted. Mapping continuous numerical values to pitch, time information to rhythmic patterning, key events to loudness and data concerning height to pitch height have some support by evaluation [14], but in general, few studies have systematically evaluated the efficiency of different sonifications [15].

1.2 Training in Manual Wheelchair Operation

According to Goosey-Tolfrey [16] the interaction between rider and wheelchair is complex and still not well understood, and overall performance depends on the ergonomics of wheelchair, the individual physical capacity, and driving technique. Furthermore, ergonomic adjustments and driving techniques vary between individuals, and is largely based on individual preference and self-learned adjustments. Modern physiotherapeutic methods for training include strategies for minimizing effort and maximizing control, derived from studies of mechanics of the wheelchair and bio-mechanics in the wheelchair-user configuration [7, 17], but they still remain somewhat imprecise. There is some consensus on methodological practices when teaching technique or evaluating capacity as to using generic exercises in testing [18, 19].

2. METHOD

2.1 Sonification System

The system consists of an *OptiTrack Prime 41* motion capture (MoCap), a computer running Pure Data, and eight (8) speakers set up in an octagon shape. [1] Spatialized sound is produced based on pre-defined movement parameters. An everyday-life wheelchair was equipped with three passive MoCap markers, one on each drive wheel axis and one on the right side of the wheelchair footrest. Four movement parameters were calculated from the MoCap, related to heading (direction), turning, speed, and tilting. [2] Table 1 shows the parameters and their mapping to the sonification models with link to sound examples.

Three different sonifications, or sound models, are named hereafter SM1–SM3. The SM1 is synthesized using additive synthesis of seven sine wave oscillators, with static proportion between oscillators. SM2 is synthesized using two mixed and continuously looped pre-recorded sounds. SM3 is synthesized using subtractive synthesis of white noise with four bandpass filters having center frequencies at odd harmonic intervals to produce a flute like sound. All sound models are spatialized to the heading using vector based amplitude panning, and they are used in two combinations: SM1+SM3 for sonifying turning and SM2+SM3 for sonifying speed.

2.2 Experiment

Four experiments sessions were conducted in this work, during which participants performed exercises derived from the methodologies of the book “Drivkraft” [7] and from works by Vereecken [19] and Inkpen [18]. The exercises were chosen to include basic movements at intermediate and advanced levels. After the exercise session, participants answered questions in a semi-structured interview.

For experiment session 1, one expert manual wheelchair user was recruited from a pre-study. The participant, male age=45, had actively used a manual wheelchair for 25 years and played wheelchair basketball at elite level. The participant reported having never heard a sonification before. For experiment sessions 2–4, novice subjects without disabilities were recruited (N=1, 1, and 4 respectively).

[1] Additionally, a plate with four pressure sensors was placed under the seat of the wheelchair. This was however not working optimally and has been omitted from the study as it did not affect the results.

[2] One parameter, related to leaning, was calculated from pressure sensors, but omitted as stated above.

Parameter	Description	Sonification	Mapping	Sound*
Heading	Direction of wheelchair	Spatialization	Panning	
Turning	Wheels rotational difference	SM1: Drone	Linear; fast turning –> high frequency	
Speed	Wheels rotational speed sum	SM2: Loop	Linear; high speed –> high frequency	
Tilting	Footrest vertical displacement	SM3: Flute	Linear; high position –> high frequency	

* sound examples: `https://soundcloud.com/user-61759848-282584838/sets/sound-examples-smc2016/s-wNAvs`

Table 1. Physical movement parameters extracted from motion capture and sensors, and their mapping to a sonification model. Links to sound examples are in the right column.

Exercise 1 drive in a figure-of-eight shape between cones separated by $1.5m$

Exercise 2 drive back and forth between cones separated by $5.5m$, alternate turns clockwise $180°$ and counter-clockwise $180°$

Exercise 3 drive in back wheel balance $3m$ across the room, passing two sets of cone-gates separated by $1m$

Exercise 4 propel forward and without further pushing the drive rings turn $120°$ around a cone and steer towards a goal cone

The four experiment sessions followed a similar procedure except for some changes: In session 1, the expert participant did an alternative version of exercise 3, driving backwards in back wheel balance; Session 1–3 were performed individually, while session 4 was a focus group study with additional discussion and demonstrations.

The participants were told what the technical components of the sonification system were and that turning and tilting [3] will produce auditory feedback. The participants were not told of the spatialization of sound nor of the variant with speed mapped to sound. The participants were then instructed too freely operate the wheelchair to gain familiarity with the system. Before proceeding to the exercises the participants were told to perform each exercise with speed and precision. It was emphasized that the goal of the exercises were primarily good precision, and secondary high speed. The participants were told to notify when satisfied with both precision and speed to proceed to the next exercise. There was no time limit imposed on the participants. After each exercise the participants were briefly reminded to execute the exercises with speed while focusing on precision.

Each exercise was explained to the participants before they performed it without any auditory feedback. After reporting being satisfied with the exercise execution the **turning sonification (SM1+SM3)** was activated and the participants continued to do the exercise until satisfied. Before moving to the next exercise the participants were allowed to question or comment. Exercises were performed successively in this manner. After completing all four exercises the sonification was switched to **speed sonification (SM2+SM3)** and the participants were told of the change and instructed to perform exercise 4 once more. Again, the participants were told to finish when satisfied with both speed and precision. After finishing all of the exercises the participants were asked a set of questions in a semi-structured interview.

[3] and in sessions 1 and 2 even leaning, but due to physiological and technical issues, this was omitted.

Figure 1. Picture shows a participant during execution of one of the exercises in the experiment.

3. RESULTS

Due to the selection of participants, the four experiment sessions were quite different even though the protocol was similar. In session 1, the participant was an expert rider, in sessions 2 and 3 the participants were inexperienced, while session 4 included group activities with an inexperienced focus group. In the following, the results will be presented accordingly.

3.1 Observations

The expert wheelchair rider, Male45, performed all exercises with ease and according to the instructions using less than 20 minutes to complete them. The participant did not express any difficulties with understanding the exercises. Exercises 1 and 2 were performed with the least variation, the participant did not ask any further questions after having been instructed. In exercise 3 the participant tried more variations as to how the exercise could be performed. The ultimate movement pattern, before proceeding to exercise 4, meant taking short and powerful strokes during back wheel balance with large variation in tilt. In exercise 4 the participant tried many variations on the turning. Since some room was given as to the approach to the cones, the instruction did not say to propel in a straight line, the par-

ticipant tried variations on the angle of the turn.

The novice participants executed the exercises in approximately 5 minutes per person and exercise. The participants in experiment sessions 2 and 3, Male28 and Male22, were able to complete all of the four exercises, although exercises 3 and 4 only after some practice. None of the participants in experiment session 4 could complete exercise 3 and two of the participants could not complete exercise 4 while the other two could.

3.2 Interview Summary Expert Participant

The participant reported hearing the auditory feedback clearly and that the sounds were relatively pleasant. When asked about which movements that generated sound, he said that he understood and added that the tipping sound (VDF-SM3) differed from the rest in being distinctly connected to the movement and that the connection between movement and sound was obvious. When asked about the spatialization of sound he had to think for a bit and then said that he felt like he noticed but that it wasn't apparent. The participant felt that the auditory feedback was well synchronized with his movement, especially the tipping sound. He said he did not know if the reason was that the movement was so distinct or that the sound was distinct. He said that the sonification of wheelchair speed (SRS-SM2) also felt distinctly connected to his movements and that the sound produced was *"obvious"*. He added that the other movements did not feel as distinctly synchronized to his movements.

The sonification of turning (SM1) was reported as less obvious by the participant. He said he could notice the sounds when turning but not in the distinct way as for the tipping sound, *"the feeling was that when I tipped [the wheelchair] it was distinct, it felt like playing an instrument. It didn't feel like that when I was turning."* When asked about whether it was the wheelchair or himself producing sound he replied that it was more the wheelchair producing sound than himself. When asked about the consistency in sound for the same movement the participant replied that it sounded the same way when he tipped the wheelchair but that the other sounds were more diffuse. He explained that when he made a turn it did not feel like he could produce a specific sound from a specific kind of turn. When further asked about the sound produced when turning he said that he heard approximately the same sound for the same performed turning. Specifically in the case of exercise 4, the participant reported not hearing a difference in the sound between consecutive similar executions of the exercise, but some difference between different executions. When asked if the auditory feedback could help him find his way back to or reproduce the same movement he replied that it did not, but that the reason probably was that the exercises were too easy for him and that if there was a more complex movement pattern that he had to execute it was more likely that the auditory feedback would help him.

The movement he thought was the hardest to execute was propelling backwards in back wheel balance, in exercise 3, and he said that he would have liked to have more help from the feedback in how straight he was going backwards. The participant also said that during the execution of that exercise he would have been helped more by the tipping sound if it gave information on the acuteness of the tilt, he explicitly requested that the sonification would give information on how close he was to tipping over. When asked if auditory feedback on operation can be meaningful the participant replied that he thought it is meaningful, but that in the context of performing these particular exercises, which were not challenging to him, the sonification system did not give him any additional meaning.

3.3 Interview and Focus Group Summary Novice Users

All participants thought that the sonification was clearly audible. Two of the participants expressed that they occasionally found the feedback to be annoying. All participants understood what movements generated sound with the participant in experiment 3, Male22, pointing out that the initial explanation from the test leader had not been necessary since the mapping was evident after some initial maneuvering.

When asked about the synchronization of movements with sound all participants said they thought it was well synchronized apart from a few short malfunctions, mostly related to the participants moving outside of the area where all rigid body markers are visible to the motion capture cameras. Male28 expressed that the sounds felt well synchronized but that there was information missing, saying, *"there were lots of movements that did not make any sound"*. The group concluded that the difficulty of the exercises made it hard to focus on the sound and thus that some of the experience of coherence in the relation between movement and sound could not be observed. Male28 stated that it was hard to listen to all of the feedback while maneuvering. When asked whether it is the participant or the wheelchair generating sound, Male28 answered it was the wheelchair, Male22 that it was a combination and in the group all participants said they thought it was a cooperation; one participant elaborated and said, *"you put order, the wheelchair is the medium"* and another participant said, *"I make the sound through the wheelchair"*.

Male28 said he thought he would need some more time to practice with the system to tell if the sonification could help him reproduce movements. He continued saying that the sonification of speed, SM2, made it obvious for him how much speed he maintained in exercise 4 and that it helped him in hearing how well he performed between runs. He also expressed that the sonification of tilt, SM3, helped him the most of all the feedback. He said that it helped him to keep an even and good back wheel balance. He also said that the sonification of turning, SM1, did not help him in executing the exercises. Male22 said he thought that it sounded the same every time you performed the same movement, *"when it felt the same, it sounded the same"*, and added that it was the most obvious for the sonification of speed, SM2, in exercise 4. Male28 said he thought it sounded similar every time you performed the same movement. Male22 concluded that the feedback did not help him to reproduce movements in the short time that

the experiment lasted for, but that the sonification of tilt clearly helped him to control the movement.

One participant in the group said she thought that the feedback, in general, does help to control movement. Another participant in the group suggested that trying to repeat a sound they were played beforehand might be easier in reproducing movement because you can focus on the sound only first and then on the sound and movement combination. Male28 made a similar statement.

The spatialization of sound was clearly audible to all of the participants in experiment 4. Male28 did not feel like he could clearly hear that all of the auditory feedback was spatialized, however, after the interview he got to listen when the test leader maneuvered the wheelchair and he then reported that the spatialization was obvious. Male22 said he did not really think about it during the experiment but that it felt like the sound followed him when he turned. He also reported that the spatialization was obvious after listening when the test leader maneuvered the wheelchair.

4. DISCUSSION

If the auditory feedback is experienced as being well synchronized with the wheelchair riders movements and the system as a whole is deemed comprehensible by the user, then there might be reason to believe that the approach is usable in a training context. It is however important to note that this depends on the level of difficulty of the exercises performed. The mapping may be well suited for a specific type of movement but if this movement covers only an intermediate part of the learning process then the system can only be used during that learning window.

The perception of synchronization between movement and sound in the system seems to have been strong. The expert wheelchair rider, Male45, considered all of the feedback to be well synchronized with his movements, however the only part of the sonification that he thought was distinct was the sonification of tilt. It is possible that the reason for that was that the sound of that feedback (SM3) was more distinct. It is also possible that the feeling of getting distinct feedback for that movement was due to the nature of the movement itself. Tilting has a definite start and stop and the feeling of shift of balance is also obvious to Male45, who concluded that the system as is would not help him control movements in the exercises given during the experiment. It seems likely that this is due to the fact that, as Male45 pointed out, the exercises were too easy for him. Some of the exercises that could have been more challenging to him would have been hard to perform in the experiment room because of its relatively small size. The part of the experiment where he had to perform back wheel balance was the only part he considered to be challenging to him. It is interesting that this is also the sonification part that he found to be most useful and where the feedback was most *"distinct"*. So the fact that some novices expressed that they had a feeling of the feedback helping in controlling movement other than tilting could point to that the individual level of difficulty affects the perception of the feedback as help in controlling movement.

In exercise 4, the sonification of turning was tested as well as the sonification of propulsion speed. In that specific context there seemed like the sonification of speed was more successful. While turning sonification could give feedback on the turning characteristics the speed sonification gave a more direct measure on how well the exercise was executed. The participants could clearly hear how much speed was maintained after turning and compare between consecutive executions of the exercise. These two different parameters for sonification then differs not only in its measure of turning performance but also in the time span through which the participant gains insight. The sonification of speed gives a direct measure on how well the exercise was performed. The sonification of turning gives information during the turning operation and it is required of the participant to listen and learn the difference between consecutive runs of the exercise until, after some training, the participant can utilize the feedback for performance improvement. The latter effect could not be observed during these experiments. The fact that the sonification of speed was deemed *"distinct"* by Male45 and useful by both Male28 and Male22 must be seen from the perspective of the above. Though, it is possible that part of the difference is due to the continuity in the sound of the SM1 and SM2, since speed sonification will give continuos feedback while the wheelchair is above the threshold speed while turning sonification will only give feedback during the operation of turning the wheelchair. It could be interesting to remove the threshold for turning sonification. This would mean getting feedback from noise in the system or from only slight turning but would also mean that the participant would get more continuous feedback.

The reason that the tilting sonification was described as more precise and distinct could be that the SM3 sounds more distinct. So it is possible that a more distinct feeling could be acquired from the other parameters mapped if the sound model was changed. The reason for choosing the specific sound models were to produce sounds that were pleasant and that differed from the each other enough to be clearly separated. It is possible that even though the intention was to separate the different parts of the feedback the choice of sound models instead impaired the comprehensibility of the system.

When talking about whether it was the wheelchair or the person generating sound there seemed to be more confidence to say that it was the wheelchair when the person had more skill. All participants concluded, however, that the wheelchair was the sound producing part of the sonification. Since this was apparent, the system can be considered successful in sonifying the wheelchair without sonifying the person. Thus there might be reason to believe that since the sonfication system did not to any large extent help in controlling movements the approach of sonifying the wheelchair while not sonifying the deviation from a good wheelchair movement was not successful for training. However, further research could show an effect on training after multiple sessions with the system.

The system is in general experienced by the participants to give synchronized feedback and in some cases precise and useful feedback that help movement execution. The

approach of sonifying wheelchair movement parameters directly and not in relation to a perfect movement execution requires that the feedback is comprehensible, information rich and well synchronized with the participants movements. If the system is rich enough on information and comprehensible is something that would have to be shown by participants clearly improving their performance after further training with the system. Furthermore, to really validate the system, a number of performance qualities (e.g. movement jitter) and performance metrics (e.g. task completion time) need to be tracked and analysed.

5. CONCLUSION

Sonification of manual wheelchair movements, with direct mapping from movement parameters, does not seem to produce feedback that is meaningful for all types of manual wheelchair operation. It may be used in specific time windows of training and for some movements that are discrete in their nature, as back wheel balance. Participants in this study did however report a good connection between sound and movement when utilizing the system developed which makes it plausible that there is applicability for the system to be used in a training context. It seems likely that for sonification of manual wheelchair operation to be successful it is a good approach to sonify the deviation from a perfect execution rather than direct sonification.

6. REFERENCES

[1] A. M. Liberman and I. G. Mattingly, "The motor theory of speech perception revised," *Cognition*, vol. 21, no. 1, pp. 1–36, 1985.

[2] M. Wilson and G. Knoblich, "The case for motor involvement in perceiving conspecifics," *Psychological bulletin*, vol. 131, no. 3, pp. 460–473, 2005.

[3] M. M. Wanderley and P. Depalle, "Gestural control of sound synthesis," *Proceedings of the IEEE*, vol. 92, no. 4, pp. 632–644, 2004.

[4] A. R. Jensenius, "ACTION – SOUND. Developing methods and tools to study music-related body movement," *Musicology*, vol. 84, no. 2, p. 275, 2007.

[5] T. Hermann and A. Hunt, "An introduction to interactive sonification," *IEEE Multimedia*, vol. 12, no. 2, pp. 20–24, 2005.

[6] A. O. Effenberg, "Movement sonification: Effects on perception and action," *IEEE Multimedia*, vol. 12, no. 2, pp. 53–59, 2005.

[7] Å. Norsten, *Drivkraft: körergonomi, rullstolsteknik & metodik*. Vällingby: Hjälpmedelsinstitutet, 2001.

[8] T. Hermann, "Taxonomy and definitions for soinification and auditory display," in *Proceedings of the 14th International Conference on Auditory Display*, 2008, pp. 1–8.

[9] N. Schaffert, K. Mattes, and A. O. Effenberg, "A sound design for acoustic feedback in elite sports," *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 5954 LNCS, pp. 143–165, 2010.

[10] R. Sigrist, G. Rauter, L. Marchal-Crespo, R. Riener, and P. Wolf, "Sonification and haptic feedback in addition to visual feedback enhances complex motor task learning," *Experimental Brain Research*, vol. 233, no. 3, pp. 909–925, 2014.

[11] T. Hermann, O. Höner, and H. Ritter, "AcouMotion – an interactive sonification system for acoustic motion control," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 3881 LNAI, 2006, pp. 312–323.

[12] L. Chiari, M. Dozza, A. Cappello, F. B. Horak, V. Macellari, and D. Giansanti, "Audio-biofeedback for balance improvement: An accelerometry-based system," *IEEE Transactions on Biomedical Engineering*, vol. 52, no. 12, pp. 2108–2111, 2005.

[13] K. Vogt, D. Pirrò, I. Kobenz, R. Höldrich, and G. Eckel, "PhysioSonic – evaluated movement sonification as auditory feedback in physiotherapy," *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 5954 LNCS, pp. 103–120, 2010.

[14] R. Sigrist, G. Rauter, R. Riener, and P. Wolf, "Augmented visual, auditory, haptic, and multimodal feedback in motor learning: a review," *Psychonomic bulletin & review*, vol. 20, no. 1, pp. 21–53, 2013.

[15] G. Dubus and R. Bresin, "A systematic review of mapping strategies for the sonification of physical quantities," *PLoS ONE*, vol. 8, no. 12, 2013.

[16] V. L. Goosey-Tolfrey, "Supporting the paralympic athlete: Focus on wheeled sports," *Assistive Technology Research Series*, vol. 26, no. 26, pp. 385–387, 2010.

[17] B. S. Mason, L. H. V. Van Der Woude, and V. L. Goosey-Tolfrey, "The ergonomics of wheelchair configuration for optimal performance in the wheelchair court sports," pp. 23–38, 2013.

[18] P. Inkpen, K. Parker, and R. L. Kirby, "Manual wheelchair skills capacity versus performance," *Archives of Physical Medicine and Rehabilitation*, vol. 93, no. 6, pp. 1009–1013, 2012.

[19] M. Vereecken, G. Vanderstraeten, and S. Ilsbroukx, "From "wheelchair circuit" to "wheelchair assessment instrument for people with multiple sclerosis": reliability and validity analysis of a test to assess driving skills in manual wheelchair users with multiple sclerosis," *Archives of physical medicine and rehabilitation*, vol. 93, no. 6, pp. 1052–8, Jun. 2012.

THE STATE OF THE ART ON THE EDUCATIONAL SOFTWARE TOOLS FOR ELECTROACOUSTIC COMPOSITION

Alessandro Anatrini

Hochschule für Musik und Theater Hamburg
al.anatrini@gmail.com

ABSTRACT

In the past twenty years technological development has led to an increasing interest in the employment of the information and communication technology (ICT) in music education. Research still indicates that most music teachers use technology to facilitate working in traditional composing contexts, such as score writing or MIDI keyboard sequencing, revealing a common and conservative conception of ICT as mere "toolkit" with limited application. Despite this, the exploration of the electroacoustic practices and their techniques, that are at the core of sound-based (as opposed to note-based) musical practices, have led to valuable composition projects thanks to pieces of software specifically created for educational purposes such as DSP by NOTAM and Compose with Sounds among others.

In this paper I will first give a short overview of the significant premises for an effective curriculum for middle and secondary education that can authentically include the electroacoustic composition, then I will summarize the state of the art in the development of the most significant educational software packages pointing to possible future developments.

1. INTRODUCTION

As we know from the last published PISA report (2012) the use of ICT positively supports results in core disciplines: in those countries where arts education is prioritized, students achieve better results in disciplines such as mathematics and physics.
Unfortunately, the effects of ICT tools employed in music education is under-researched (Rudi, 2013). It is a fact that studies in this area are mostly oriented towards note-based music and the tools that fit this approach. Despite this situation we are able to define the advantages created by the employment of electroacoustic practice in this context in the following fashion:

- The possibility of direct manipulation of source material provides immediate feedback and implicit encouragement.
- Allowing the use of synthesized sounds as well as a student's own recordings, is not merely an act of encouragement where students provide their to own material, but as it has already been demonstrated (Harter, 1990), they will most likely be inclined to develop a long term interest in those activities that directly concern their personal and social identity, and more generally their emerging concepts of "self".
- Working with musical signal processing technology that deals with phenomena in the sounds themselves, leads to various degrees of awareness in the intrinsic properties of sound and their role in wider musical structures. In other words, a conscious sense of the musical form is developed through practice.
- Digital tools through their very nature address cross-disciplinary projects in the broadest sense.

Nonetheless all of this does not mean that all pedagogical aspects of musical ICT can be defined as appropriate. It is crucial to pay attention to those critical issues of music curriculum development in order to avoid an outcome whereby a significant art form loses its pedagogical potential.

2. A CLEAR PERSPECTIVE

2.1. Basic assumptions

The development of virtuosic electroacoustic practice within music education curricula has been slowed down by two phenomena: Firstly by the prevailing stereotypes and by weak conceptual elaborations, and secondly by the lack of a strong link between the processing of contents put into practice by musicology and the pedagogical concepts of music. Actually, in most cases, technology's role is conceived exclusively in relation to a traditional music curriculum. This approach leads to a misunderstanding where composition based on sounds is seen as separate from "serious" composition based on notes (Martin, 2010).
We can better understand what causes this misinterpretation referring to two competing paradigms borrowed

from music pedagogy, representing two contrasting "*Weltanschauungen*".
Martin (2010) and others have already debated the argument, therefore the scope of this article will be limited to the key statements in surrounding the development of a conceptual framework that details why electroacoustic practice can be authentically pursued thanks to educational software.

2.2. A critique of "Good Taste"

The paradigm labeled as "aesthetics" traces its origins to late eighteenth and early nineteenth century romantic idealism and has been the predominant conceptual basis for music pedagogy (Reimer, 1988 and others), and consequently in education, since the middle of the twentieth century (McCarthy and Goble, 2005). The focus of this paradigm is a definition of music comprising "aesthetic or expressive elements" which according to Reimer (1988, p.52) can be defined as "rhythm, melody, harmony, tone colour, texture and form". Whether, musical activities in the classroom involve making or listening, this becomes aimed at understanding and developing these supposed intrinsic properties of musical works. In order to pursue these kinds of educational goals it becomes crucial to choose good examples of music, mostly Western art music, that supposedly have those universal qualities transcending all temporal, spatial, cultural, and social contexts. Since the education that is derived from these underlying assumptions is geared to develop sensitivity for the appreciation of musical works rather than the capacity for making music. The ideal student that should be trained, assuming that an adequate training can take place, is an expert connoisseur with a "Good taste" for a specific kind of music. Due to its nature, this approach leads to the imposition on the classroom of the aforementioned exercises instead of authentic activities, and therefore ultimately tends to fail to sufficiently involve students in music because it is not perceived as really meaningful by the students themselves.

2.3. The turning point

The focus of the praxial paradigm (Elliott, 1995) concerns practices. The emphasis shifts from the "musical objects" to the practices that produce those objects, from an ideal realm of universals to a web of social, cultural, economic and political attributes inside of which these practices take place. With these premises it becomes clear how musical meaning is in no small part inherent in the context in which the sound is produced instead of being an intrinsic value of the sound itself as the aesthetic perspective claims. The music is conceived as something people do, and therefore the different declinations of music-making such as composing, arranging, performing and improvising are the goals of an education based on this perspective. The curriculum becomes a *practicum* with the aim to approximate authentic music cultures not to duplicate them, and in which "rich and challenging music-making projects in classroom situations […] are deliberately organized as close parallels to true musical practices" (Elliott, 1995, p. 261). Within this framework a more effective and authentic learning environment can be accomplished for at least two reasons. First, the fact that students can learn through activities and problem-based situations rather than listening, makes the subject more accessible, better fitting the needs of pre-and early adolescents in their phase of development, and generally guaranteeing them better training as Lonsbury (2000) observed. Second, it is a matter of authenticity: if the music is presented merely as a school subject with activities directed to understand and to appreciate "the elements of music" the students tend to be less motivated because, as research indicates (Caskey and Anfara, 2007), young adolescents are typically drawn to real-life experiences and related learning situations that are perceived as authentic and meaningful.

3. HOW TO ACTUALIZE THE PRACTICE

The nature of electroacoustic music fits the basic assumptions of the praxial paradigm and its consequences. Researches in this direction as yet has not been undertaken and consequently there is much we do not know. My personal teaching experience led me to think that the *medium* to fully actualize the above mentioned premises should mostly be comprised of pieces of software specifically created for educational purposes for at least three reasons. Firstly, it is crucial to be able to create an educational environment that integrates conceptual and practical learning. The focus on practice does not necessarily need to lead to the misunderstanding, that conceptional learning is banned from this approach leaving the student free to find his or her own way through the possibilities enabled with the software package in question. Conceptual learning should be used to understand, to address and to facilitate the practice. The advantages of this approach are double: on the one hand, we do not run the risk of providing concepts that can be interpreted as detached from practice, since we can promptly verify conceptual notions. On the other hand, the teacher can scientifically supervise the progresses of the students learning. Second, as with regards to commercial pieces of software, with their own logic, which are geared towards professional or amateur users, they do not approximate an authentic music practice, but instead create it without mediation; the aim of music education is not to educate all students for careers as professional musicians. Eventually, what is not perceived as personally meaningful to the students is doomed to fail, even if it involves a creative attempt to interest them in electroacoustic music using the "fanciest" software. It is realistic to think that products based on the educational needs of the student offer tangible benefits related to his/her personal and social life as much as needed to continue to pursue the subject after the school years.

4. THE STATE OF THE ART

4.1. Preamble

If a "perfect tool" that includes all these features does not exist in order to teach electroacoustic music in middle and secondary school classes, the risk that these endeavours become their own means to an end in the service of a

curriculum geared towards music appreciation, is just around the corner. Nevertheless, valuable projects such as DSP, E-Lab/Live8 and Compose with Sounds represent steps in the right direction and they entail not only a correct practice, but they can also be taken as benchmarks in order to formulate possible future developments.

4.1.1. Common features

The reason why these projects embody an authentic and "correct" practice, as it has been outlined above, can be seen in the desirability of some of their characteristics.

- They get closer to professional pieces of software for music composition.
- They allow the exploration of sound and its properties through advanced and complementary software tools.
- They require a limited conceptual musical training providing at the same time a multilayered interactive help system (supported only on DSP and Compose with Sounds).
- They entail a constructivist approach with student-centered (but teacher supervised) learning perspective.
- They focus on relatively easy operations and an appealing graphic user interface (GUI).

The following section illustrates the different contexts where these pieces of software come from, as well as their specific characteristics.

4.2. DSP

Even though in the mid-90s there were few educational programs or general familiarity with music technology in Norway, the Ministry of Education developed guidelines for the employment of such technology in schools. Regarding music, children were supposed to compose using technology. The context of the birth of the project was pioneering because the digital revolution was only in its early stages and computers still had strong computational limits. DSP (Digital Sound Processing) has been the first software expressly created for educational purposes which was developed and maintained by the Norwegian Network for Technology, Acoustic and Music (NOTAM) from 1996 to 2013[1].

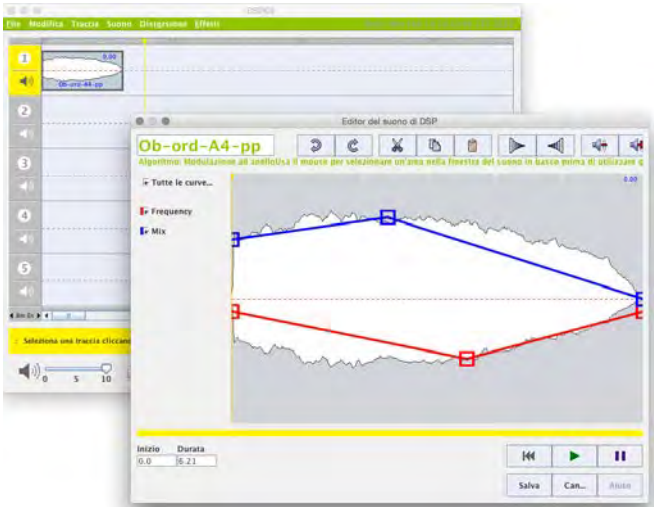


Figure 1. DSP screenshot of the mixing and editing windows.

As we can clearly see in Fig. 1 the aim of the project was to develop a "software model for education that […] resembles a professional software approach for computer composition, such as signal processing software […] and screen based mixers (e.g. ProTools)" (Rudi, 2007). The mixer window (Fig. 1, background window) is composed of five tracks and some "classic" controllers, that screen-based mixers generally have, such as: playback and pan controllers, gain slider, the possibility to insert markers and to turn each track on or off and a menu bar. Both from the "Sound", "Distortion" and "Effects" menus is possible to access to the sound processing algorithms (fig. 1, front page window) that include chorus, flanger, delay, harmonizer, filters, reverb, ring modulation, operations in the spectral domain, time stretching, granulation, scratch, as well as the possibility to generate sounds or musical structures from mathematical models; real-time processing of audio streams is not supported. Since DSP is no longer maintained it is not clear if it is still possible to connect it to the NOTAM website in order to read the tutorial texts accessible through the "Help" button in the editing window. During the past years this innovative approach to music education has enhanced the popularity of DSP thanks to music projects and workshops in Norway, Sweden, Denmark, UK, France, Portugal and Italy as it is demonstrated by the languages available for the software.

4.3. E-Lab and Live_8

The two Max-based[2] learning environments were developed at Tempo Reale, a technology-based music center for production, research and education, established in 1987 by Luciano Berio in Florence. From 1999 Berio started to promote the planning and the development of new educational activities focused on creative practices and aimed at 8 - 13 year old primary and middle school children, through the use of pieces of software specifically created for educational purposes. After an experimental phase ran at IRCAM-Centre G. Pompidou[3], since 2000 the project has been developed and spread in many Italian

[1] Last available version is updated to february 2013, http://archive.notam02.no/DSP02/

[2] Max is a visual programming language for music and multimedia developed by Cycling'74, http://cycling74.com.

[3] *Atelier Jeunes* created by T. Coduys and J. Baboni-Schilingi.

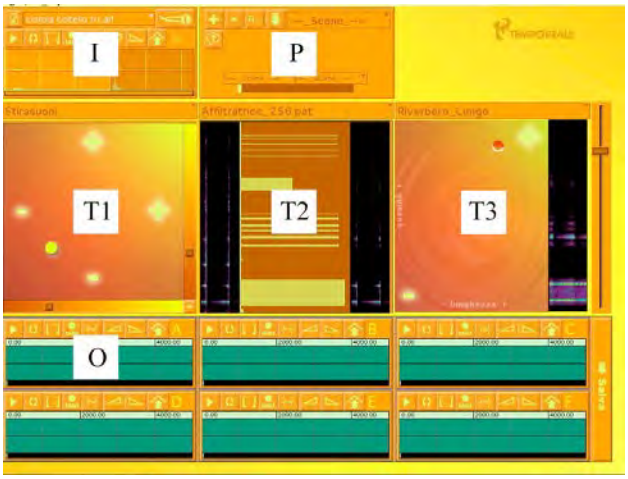


Figure 2. E-Lab screenshot.

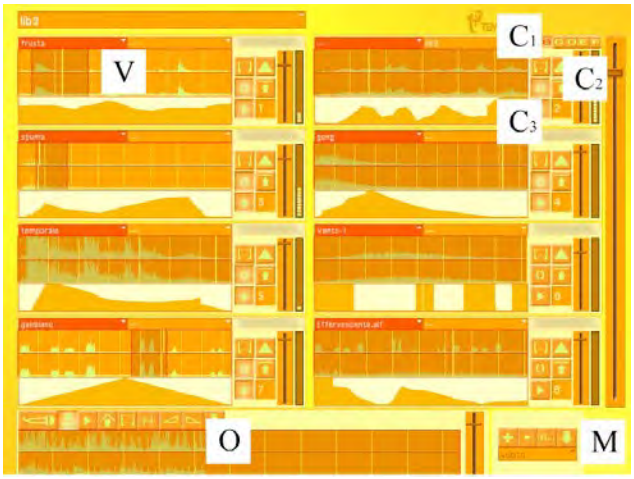


Figure 3. Live_8 screenshot.

cities, with the participation of more than 1700 students, and culminated between 2005 and 2007 in the realization, among others, of the final version of E-Lab, Live_8 and the Gamelan_01[4] project. Since 2009 these pieces of software were no longer maintained.

The aim of E-Lab is the discovery of the morphological qualities of sound through the comparison between original and edited sounds, as well as the development of the skills to handle the sounds' morphology (Luca, 2009). Unlike DSP, the *concept* of E-Lab is not borrowed from screen-based mixer. Rather it is inspired by a matrix editing environment with a changeable set, that allows complex manipulations through the combination of easy transformations. It consists of an input area (Fig. 2, letter I) where the waveform is displayed and includes: playback controllers, the possibility to load stored sounds as well as to record audio streams and fade-in fade-out edit modes. Close to this area it is possible to store parameters in order to recall different sets (Fig. 2, letter P). The letters T correspond to the editing area where is possible to load up to three different modules. Eight modules are available:

- "Stirasuoni" for frequency and time domain transpositions.
- "Congelatore" based on a freezing technique.
- "Eco" to create rhythmic and melodic structures, based on a delay-harmonizer technique.
- "Massificatore" to create sound masses through a polyphonic reading technique.
- "Naviga-Suono" based on granular synthesis in order to explore the sound in any direction and at any speed.
- "Affiltratrice" a 256 band spectral domain filter.
- "Incrocia-Suoni" for the blending of two sound sources, based on the vocoder technique.
- "Riverbero" a reverb effect.

The letter O corresponds to the output area where it is possible to display and save the outcome of the edit modules. This kind of approach to the learning environment ensures at least two advantages. First, reversibility: if the sound is processed through, for example, three modules (a, b, c) is possible to collect the result not only at the end of the process but also at an intermediate stage (e.g. between a and b). Second, visibility: for each and every transformation the result is represented through a real time graphical representation of the waveform in order to facilitate the consciousness of the alteration of the morphology of sound.
E-Lab was conceived to lead the students through the discovery of the physical qualities of sounds, whereas Live_8 is oriented to introduce them to become aware of the issues concerning the handling of the musical form. Live_8 is a *live* environment whereby is possible to edit, mix and launch up to eight different stored samples at the same time. The GUI is based on two devices: a keyboard-like area (Fig. 3 letters C1, C2, C3) with playback, gain and loop controllers, and a display for the representation of the waveform in time domain. Furthermore, it is possible to observe and to edit again the outcome of the mix (Fig. 3 letter O) as well as to save the configuration of the device (Fig. 3 letter M).

4.4. Compose with Sounds (CWS)

The software is a part of a wider initiative starting from the EARS pedagogical site[5] (Landy, 2007, 2012). "*Compose with Sounds* is an EU Culture program-funded software development project (2011-13) initiated by the Music, Technology and Innovation Centre at De Monfort University (UK). It involved partners in France (ina-GRM), Germany (ZKM) and Norway (NOTAM) as well as associate partners in Greece (EPHMEE/Ionian University in Corfu) and Portugal (Miso Music, Lisbon). The goal of the project was to lower the threshold for young composers who want to explore sound-based music and

[4] *Gamelan_01* involved an orchestra of 80 children of 9 y.o. playing a real time electroacoustic composition developed during the preparatory workshops. It took place in 2007 at Teatro La Pergola, Florence. An excerpt of the performance is available at http://vimeo.com/138252774.

[5] The ElectroAcoustic Resource Site (EARS) http://ears2.dmu.ac.uk is a dynamic eLearning site with a wide variety of hypermedia examples. It has been established to provide resources for those wishing to conduct research in the area of electroacoustic music studies as well as to introduce people of all ages, but especially those starting secondary education and their teachers, to the world of making music with sounds.

music composition, and to provide schools and individuals with an educational tool that encourages creativity, cross-cultural dialogue and the sharing of ideas and results"[6].

CWS represents a necessary development of the basic assumption of the DSP project in the context of a more technologically mature situation as once increased processing power at cheaper price became available. Although the reference model of the screen-based mixer remains the same, and both projects do not support real-time processing of audio streams, CWS presents some desirable innovations and enhancements compared with its predecessor. The new GUI allows users an interaction with sound at five different levels:

- Sequencer: (Fig. 4 letter S) is the main window and the environment to handle and arrange the samples. It is composed of the sound library viewer, the time bar just below, an arrangement area and a control panel.
- Sound cards: the pictures displayed in the interactive sound library viewer window (Fig. 4 letter Sc) represent the sounds that can be used in a composition just by dragging them to the arrange area. It is possible to expand or build new libraries adding students' own sounds.
- Edit modes: the controls (Fig. 4 letters E1, 2, 3) allows the user to change the way the sounds are displayed in the arrange area. E1 card mode, E2 waveform mode and E3 automation mode.
- Automation: the user can access the automation mode through the button (E3) or through the tiny picture above any samples in the arrange area. The mode allows the user to describe a pattern, with a break point curve (Fig. 4 letter A), for any parameter of the software.
- Manipulation window: it appears once the top of a card in the sequencer is double-clicked. The window is for adding and manipulating effects[7], it contains a visualizer as well, which helps the student understanding the effect he or she is using (fig.5). All the tutorial texts and videos are linked to the pedagogical ElectroAcoustic Resource Site.

Figure 4. Compose with Sounds sequencer screenshot.

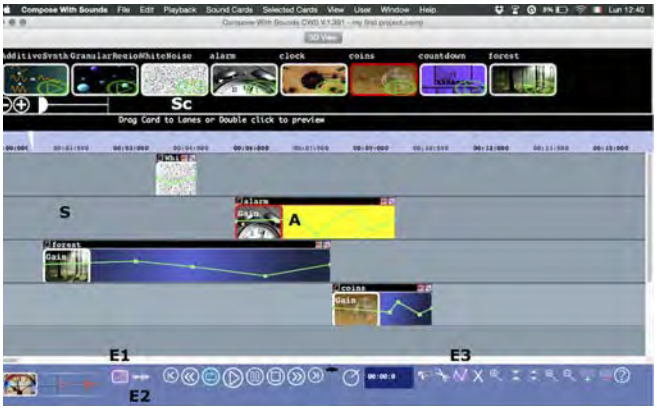


Figure 5. Delay viewer on the manipulation window.

A number of schools in the six involved countries took part in this project. This has contributed to the further development of the software and teaching methodologies between 2011 and 2013. In order to secure a link with professional communities, composers worked alongside pupils resulting in the creation of a substantial number of works that were presented in concerts across Europe forming one of the key project outcomes[8]. Since 2013 the software is no longer maintained.

5. OBSERVATIONS

Despite different approaches to the critical issues of the concepts of the software these projects have granted some shared benefits to thousands of students involved in nine different countries. During the activities the majority of them developed a responsible and mindful attitude regarding their results, an increased critical sensibility and self-esteem, students with learning and relational difficulties also found the opportunity to express themselves positively emphasizing their characteristics (Luca, 2008).

Although these projects are no longer maintained (except the case of the EARS2 site), they undoubtedly have contributed to the creation of a school in Europe. In view of these experiences I suggest the following guidelines concerning future desirable developments.

- More than 30000 music apps are available on the *Apple App Store*, more than 3000 on *Google Play*, and making music on smart devices has been more than a simple novelty as already demonstrated by some excellent music apps like Audiobus, Beatsurfing, Mitosynth, SECTOR, SunVox, TC-11 and many more (Krebs, 2014, 2012). An up-to-date technological perspective should take into consideration the creation of educational apps

[6] From the Compose With Sounds website: http://cws.dmu.ac.uk/EN/11.

[7] A list of all effects and tutorial videos is available at http://cws.dmu.ac.uk/cwshelp/help_e/effects.html.

[8] A selection of works composed by the students is available at http://cws.dmu.ac.uk/EN/10.

for smartphones and tablets instead of pieces of software for laptops and personal computers[9]. This fact presents more than one benefit. First, the software experience is enhanced not only through the touch screen, but also thanks to the possibility to use the sensors that are standard in such devices such as microphones, cameras, gyroscopes and compasses. Second, the learning environment is no longer constrained to the front of a screen and instead becomes *mobile*, stimulating the student, to explore the relation between movement and sound. Third, many students already own a smart device.

- The screen-based mixer approach should be abandoned in favor of a more informed perspective pointed towards the already existing commercial apps and more generally to the critical issues of the design-based researches (Aigner, 2015).
- A shared composition practice should be enhanced and based on standard orchestral practice, where each musician contributes to the creation of the performance by playing their part. The notable advantages of making music together are well known and cannot be discussed here, in any case apps or pieces of software that allow the students to work in group on the same composition in real time do not exist yet. This perspective should be carefully researched.

6. CONCLUSION

Technology has revolutionized how music in schools is taught, whilst the ways in which we access and listen to music through downloads, streaming libraries and smart devices have changed our engagement with it. Even though the response of music educators is not always ambitious or pedagogical and technologically up-to-date, it is clear that new media promise a longer and more durable interest in the subject. The ways to pass down the knowledge of the educational pieces of software for electroacoustic composition into the world of the smart devices has to be carefully devised. This new path has just begun.

7. REFERENCES

1. J. Rudi, “Research on education in electroacoustic composition with children - future challenges”, in Electronic sounds in the Classroom, ZKM, Karlsruhe, 2013.
2. S. Harter, “Causes, correlates and functional role of global selfworth: A life-span perspective”, in Perceptions of competence and incompetence across the life-span, Yale University Press, 1990, pp. 67 - 98.
3. J. Martin, “Electroacoustic music in middle and secondary education: Some concerns regarding curriculum development”, in Proc. Int. Conf. Teaching electroacoustic music: Tools, Analysis, Composition (EMS2010), Shanghai, 2010, pp.1-12.
4. B. Reimer, “A philosophy of music education”, Prentice Hall, 1988.
5. M. McCarthy and S. Goble, “The praxial philosophy in historical perspective”, in Praxial music education: Reflections and dialogues, Oxford University Press, 2005, pp. 19 - 51.
6. D. J. Elliott, “Music matters: A new philosophy of music education”, Oxford University Press, 1995.
7. J. H. Lonsbury, “Understanding and appreciating the wonder years”, National Middle School Association, 2000. Available at: https://www.amle.org/portals/0/pdf/mlem/wonder_years.pdf.
8. M. M. Caskey and V. A. Anfara Jr., “Research summary: Young adolescents’ developmental characteristics”, Portland State University, 2007. Available at: http://pdxscholar.library.pdx.edu/cgi/viewcontent.cgi?article=1009&context=ci_fac.
9. J. Rudi, “Computer Music Composition for Children”, in IEEE Signal Processing Magazing, 24 (2), 2007, pp. 140 - 143.
10. S. Luca, “E-Lab, a new learning environment for developing children’s musical creativity”, in Proc. European Network of Music Educators and Researchers of Young Children (MERYC2009), Bologna, 2009, pp. 409 - 419.
11. S. Luca, “Esplorazione e creazione musicale con le nuove tecnologie” in Musica Domani n. 147, 2008, pp. 12 - 19.
12. L. Landy, “The electroacoustic resource site (EARS) approaches its next phase: going global and adressing the young, in Proc. International Computer Music Conference (ICMC2007), Copenhagen, 2007, pp. 141 - 144.
13. L. Landy, “Making Music with Sounds”, Routledge, 2012.
14. M. Krebs, “Musikinstrumente im Taschenformat: Erforschung und Anwendung der App-Musik stehen erst am Anfang” in NMZ, 2/2014, pp. 10 - 15.
15. M. Krebs, “APP-MUSIK - Musizieren mit Smartphones. Perspektiven und Potenziale einer neuen musikalischen Form” in MusikForum 1/2012 pp. 14 - 19.
16. W. Aigner, “Design-based research in music education. An approach to interlink research and the development of educational innovation”, in Proc. Int. Conf. Open Ears - Open Minds. Listening and Understanding Music (EAS2015), Rostock.

[9] Interesting projects involving mostly primary school children have already taken place in Berlin, have a look at http://app2music.de.

REVEALING THE SECRET OF "GROOVE" SINGING: ANALYSIS OF J-POP MUSIC

Masaru Arai, Tastuya Matoba
formerly at Kwansei Gakuin University
{riviera314,mtb.toya0403}@gmail.com

Mitsuyo Hashida
Soai University
hashida@soai.ac.jp

Haruhiro Katayose
Kwansei Gakuin University
katayose@kwansei.ac.jp

ABSTRACT

In music, "groove" refers to the sense of rhythmic "feel" or swing. Groove, which was originally introduced to describe the taste of a bands rhythm section, has been expanded to non-rhythmic sections and to several genres and has become a key facet of popular music. Some studies have analyzed groove by investigating the delicate beat nuances of playing the drums. However, the nature of groove that is found in continuous sound has not yet been elucidated. To describe the nature of groove, we conducted an evaluative study using a questionnaire and balance method based on signal processing for vocal melodies sung by a professional popular music vocalist. We found that the control over (voiced) consonants followed by vowels constitutes an expression that is crucial to groove in J-pop vocal melodies. The experimental results suggest that time-prolongation and pitch overshoot added to voiced consonants made listeners perceive the vowels that follow to be more accentuated, eventually enhancing listeners perceptions of groove elements in vocal melodies.

1. INTRODUCTION

The rhythm of some types of music causes listeners to tap their feet and dance. This feeling is commonly referred to as groove and has a strong affective component as well as a strong correlation with music appreciation [1]. Groove originally represented a taste for performance expression commonly found in jazz rhythm sections; it has since been established as a form of rhythmic expression found in various forms of popular music such as salsa, funk, rock, fusion and soul.

Previous studies have provided a strong consensus on the definition of groove [1,2], and some researchers have quantitatively analyzed rhythmic performances [3–8]. Okudaira et al. [3,4] analyzed onset timing and the loudness of snare drum beats and reported that a micro-difference in onset timing effectively expresses groove. Madison et al. [6] followed this finding with an observation that tempo alone cannot explain groove, and they suggested that the main physical correlate of groove is syncopation [8]. Sioros et al. examined listeners experiences of groove when exposed to synthesized musical stimuli covering a range of syncopation levels and densities of musical events [9].

Furthermore, the ideology of groove for rhythm sections has recently been expanded to continuous sound expression through melodic instruments, including vocal singing. Currently, singing voice information processing is developing considerably [10], and production software such as VOCALOID [1] is found worldwide. New technologies for vocal expression serve as methods and tools to elaborate parameters of acoustic and melodic expression such as pitch, loudness, vibrato, portamento, and joint vowels and consonants [11–14]. However, the control method for the expression of "groove" singing remains obscure and underdeveloped.

In this paper, we aimed to determine which properties of vocal singing affect groove sensation. We recorded songs with and without groove elements sung by a professional singer who can intentionally control groove expression, and we then analyzed and modified several parameters of the recordings. Section 2 describes our approach to groove analysis and general information on our analysis. In Section 3, we describe our analysis of the onset timing of voiced consonants. Section 4 describes two listening experiments that focus on the "overshoot," technique, a pitch control feature of singing.

2. GROOVE SINGING ANALYSIS APPROACH

"Groove" is a musical term related to rhythm expression. "Groove," which was originally used to refer to the nuances of a rhythm section, now refers to singing skill and especially to pop music vocalists. It is not difficult for humans to distinguish "groove" singing from "Non-groove" singing. However, the properties of singing that make us feel that vocals have "groove" have not yet been elucidated.

To address this problem, we compared singing with and without "groove" elements sung by a professional J-pop vocalist and estimated vocal properties that may affect "groove." From this procedure, we found that control over consonant preceding vowels is a crucial property of "groove" singing. As a result of this procedure, we find control of consonant preceding vowels is one of the crucial properties of "groove" singing. We then investigated the effects of the lengthening and pitch overshoot of voiced consonants on the expression of "groove" singing.

[1] http://www.vocaloid.com/en/

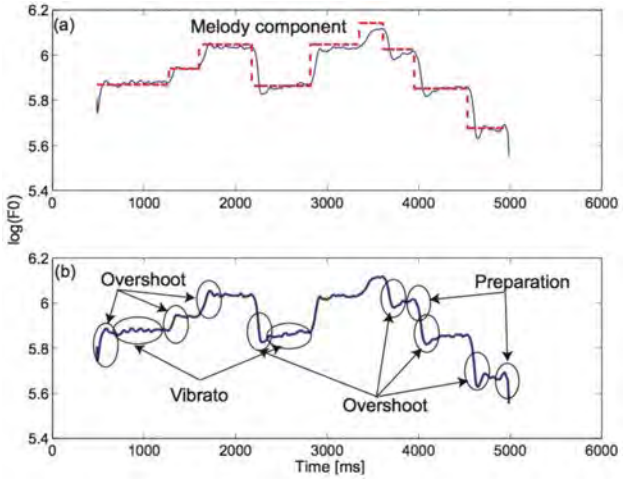


Figure 1. Musical notes and F0 dynamics from [15]

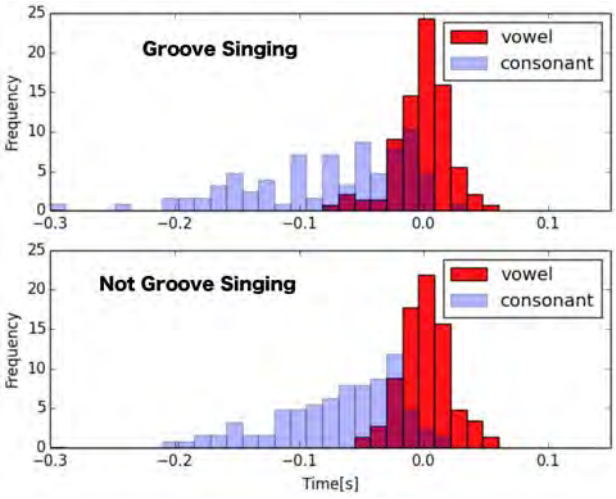


Figure 2. Histogram of deviation in consonant and vowel onset in human singing (Piece A)

2.1 Target Pieces and Recording

The music selected for the experiment needed to express groovy taste. Therefore, two type of music were selected: **Piece A** is "La La La Love Song [2] ," a middle tempo R&B and **Piece B** is "Love Rain (Koi no Ame) [3] ," a slow ballad composed by Japanese singer Toshinobu Kubota, who is known as one of the best singers of soul music in Japan.

We recorded and analyzed pieces A and B with and without "groove" elements, which were sung by a professional pop vocalist who is also an experienced vocal trainer of Japanese professional pop music vocalists.

2.2 Vocal Manipulation Tool

During the listening test, singing materials with properties that may affect "groove" are controlled. For this goal, we adopted STRAIGHT [13, 14], a tool that is capable of manipulating voice quality, timbre, pitch, speed and other attributes for research on speech and synthesis. The STRAIGHT tool can be used to control the length and pitch of any phoneme. An example of a pitch analysis using STRAIGHT is shown in Figure Figure 1 [15]. This figure shows that the naturalness of singing involves pitch transitions (e.g., overshoot, undershoot and vibrato).

2.3 Comparison of Onset Timing

Most "groove" studies have analyzed drum beats. Thus, onset of drum beat timing is regarded as a crucial property that causes listeners to experience "groove." Based on prior investigations, we first compared the onset timing of vocal melody beats between expressions with and without "groove." The effects of properties of temporal control emerging from the comparison were examined through listening experiments.

[2] https://www.youtube.com/watch?v=prfepwJ5wZE
[3] https://www.youtube.com/watch?v=KuMD-FulT5s

2.4 Analysis Focused on Voiced Consonants

Our preliminary analysis on the onset timing of vocal notes suggested that there was no significant difference between attacks among conditions with and without "grove." Rather than attacks, the analysis suggested that the lengthening of consonants preceding vowels may be a property of "groove" singing. From this finding, we conducted a more detailed analysis of controls by timing phonemes and separating vowels from consonants. Then, another experiment was conducted to verify whether the pitch overshoot of voiced consonants preceding vowels may also constitute a property of "groove" singing.

3. COMPARISON OF ONSET TIMING

This section compares the onset timing of consonants and vowels in vocals with and without "groove" elements and presents an estimation of crucial property candidates of "groove" singing.

3.1 Analysis of Vowel and Consonant Start Times

The lyrics of the target pieces used in our experiments are written in Japanese. Japanese belongs to the 'mora' language, whereby each phoneme includes a consonant and vowel. We first analyzed deviations between vowel start times and those of preceding consonants.

Figure 2 shows histograms of the start times of vowels and preceding consonants. In this figure, the start time of each phoneme is normalized, as the start time of a vowel is zero when the vowel starts at its nominal beat time. This figure suggests that vowels sung by skilled vocalists are pronounced accurately on beat time, and there are no significant differences between such features among songs sung with and without "groove" conditions. By contrast, the onset of consonants under "groove" conditions occurs earlier than under "non-groove" conditions. These results suggest that maintaining the tempo of vowels is a funda-

mental skill of rhythmic expression, and other properties are used to positively express "groove" singing.

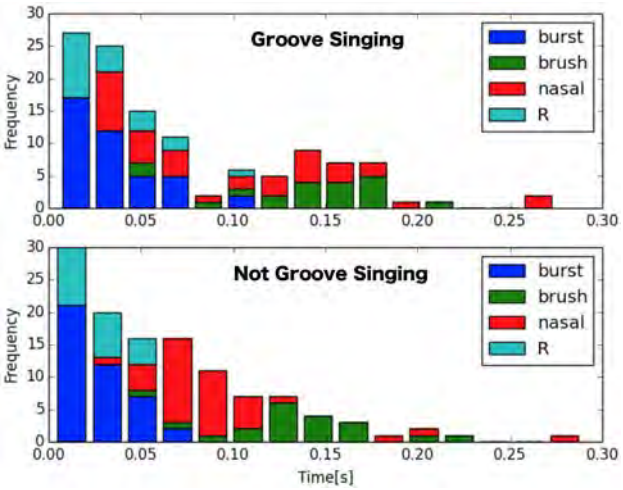


Figure 3. Histogram of consonant onset in human singing (Piece A)

Figure 3 shows histograms of consonant lengths, which are divided more precisely. This figure shows that the lengths of nasal and brush consonants are lengthened. That is, some nasal consonants are intentionally lengthened compared with other consonants in "groove" singing.

To verify that this control affects "groove" singing, we recorded "groove" and "non-groove" singing samples whereby all consonants were replaced with "m," and we then compared these samples with the original recordings.

Figure 4 shows the results of this comparison, where the same procedure as shown in Figure 2 was adopted for analysis. When comparing Figure 4 with 2, "m," is pronounced earlier than other consonants. When comparing "groove" and "non-groove" features in Figure 4, there is variance in the starting time where "m" is pronounced sooner and for a longer period in "groove" songs than in "non-groove" songs. Averages and variances of start times ahead of the nominal beat time of "m" were, 91$[ms]$ and 1,651$[ms^2]$ for "groove" songs and 76$[ms]$ and 961$[ms^2]$ for "non-groove" songs. The start time of "m" in "groove" songs was found to occur much earlier than that in "non-groove" songs ($P < 0.05$).

3.2 Listening Experiment

This section describes an experiment conducted to investigate the effects of consonant and vowel onset timing on "groove" sensation from a psychological point of view. In the experiment, participants were asked to state which song they felt exhibited more "groove," using Scheffe's paired comparison based on eight sound materials (see Table 1). Each consonant and vowel was replaced to simulate "groove" and "non-groove" conditions using STRAIGHT, as shown in Figure 5. Melodies for this part of the experiment were selected from those including conjunct motion, disjunct motion, and same-pitch transitions between two adjacent notes in Song B based on the implication and realization

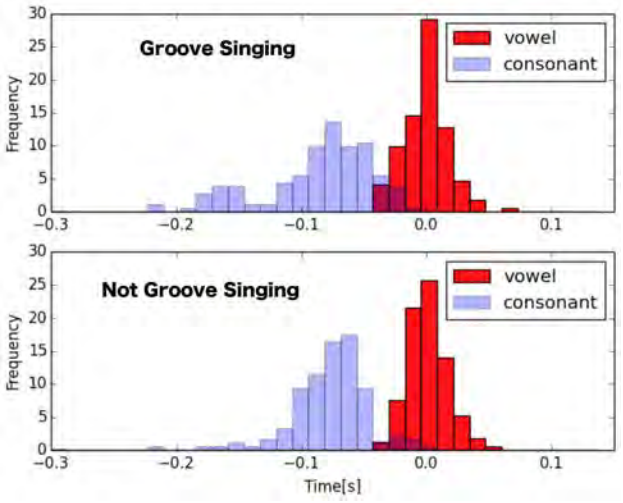


Figure 4. Histogram of deviation in vowel and consonant onset in consonant-controlled singing (Piece A). All consonants are replaced with "m."

Phrase No.	Human singing	Consonant length	Vowel onset
a	groove	groove	groove
b	groove	groove	non-groove
c	groove	non-groove	groove
d	groove	non-groove	non-groove
e	non-groove	groove	groove
f	non-groove	groove	non-groove
g	non-groove	non-groove	groove
h	non-groove	non-groove	non-groove

Table 1. Phrase listening stimuli patterns

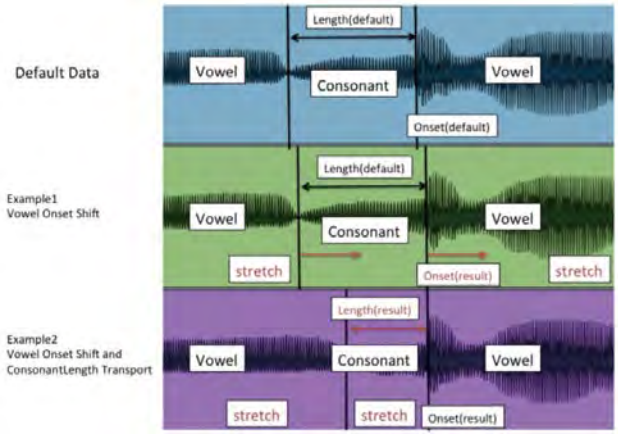


Figure 5. Manipulation of sound materials using STRAIGHT

model proposed by Eugene Narmour [16].

Thirty-six individuals participated in this experiment (male: 28, female: 8). Among them, 24 reported having experience with music. All the participants selected either the originally recorded "groove" sample or a consonant stimulus that was replaced with that of the originally recorded "groove" sample, as the "groove" stimulus. This result

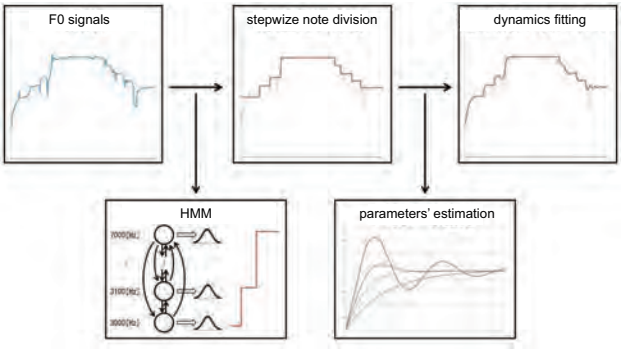


Figure 6. Procedural overview of the F0 transition analysis

suggests that listeners judge "groove" taste based on singing while listening for consonant control features.

3.3 Discussion

No major differences were found in the pronunciation of vowels in songs with and without "groove" conditions. Nevertheless, listeners participating in the experiment noted that they heard vowels with "groove" elements (i.e., consonants that are pronounced earlier and that are more accentuated). This finding suggests that auditory illusions induced by control over preceding consonants may be a central facet of "groove" singing. In the following section, we explore our more detailed experiments that focused on the pitch control of nasal consonants in "groove" singing.

4. ANALYSIS OF VOICED CONSONANT PITCH CONTROL

This section describes two experiments that were conducted to investigate effects of pitch control on groove sensation.

As noted in Section 3, we hypothesized that auditory illusion brought about by the pitch control of voiced consonants may accentuate subsequent vowels. We also found that the control of voiced consonants, and especially pitch overshoot during melodic leap progression, may be a key property of groove singing.

We thus conducted a listening experiment to investigate how the pitch control of voiced consonants of sequential and leap progression affects the loudness of subsequent vowels. We then investigated how this control may increase "groove" features of a phrase.

4.1 Voice Synthesis for the Experiments

To analyze and resynthesize the pitch control of voiced consonants, we adopted a pitch transition model that was proposed by Ohishi et al.: models human pitch control based on human physical constraints [17]. This model enables us to control natural pitch transitions including overshoots and undershoots with variables ζ and Ω, respectively, as shown in Figure 6. We analyzed an F0 sequence of voiced consonants and vowels to resynthesize songs in which the parameters changed, and we conducted a listening experiment with the resynthesized singing.

4.1.1 Note-level division of audio signals using HMM

First, the F0 sequence of singing o_{Hz} isinging is converted to logscale frequency o_{cent} as follows:

$$o_{cent} = 1200 \log_2 \frac{o_{Hz}}{440 \times 2^{\frac{3}{12}-5}} \quad (1)$$

We assume that the pitch transition follows the Elgodic Hidden-Markov Model (eHMM). In the eHMM, there are 42 states that correspond to a pitch of 700 cents from 3000 to 7000 cents. The output of each state appears as a normal distribution where the average is the frequency of pitch and the variance is 100. The self-transition possibility is 0.9, and the transition possibility for other states is $0.1/41$. Thus, the F0 sequence is finally divided into a musical notes after the route estimation of each state through a Viterbi search.

4.1.2 Least-Squares Fitting

Next, we estimate parameters of transfer functions using fitting procedures for the dynamic component.

$$H(s) = \frac{\Omega^2}{s^2 + 2\zeta\Omega s + \Omega^2} \quad (2)$$

As in pre-processing, F0 transitions of each note is normalized as follows:

1. The F0 sequence of a phrase is initialized at zero from the beginning note of a phrase,

2. If a note is NOT positioned at the beginning of a phrase, the F0 sequence of the note is subtracted from the beginning frequency and from the frequency of the preceding note.

The last step is to estimate the parameters ζ and Ω so that the sum of squares of the residuals between the F0 sequence and signal convolved with the impulse response of the transfer function is minimized.

Parameters obtained through this procedure enables us to analyze and synthesize F0 dynamics for each note.

4.2 Exp. 1: Relationship between Pitch Overshoot and Loudness

The first experiment is conducted to verify that the pitch-overshoot of voiced consonants causes listeners to feel that a note includes a louder consonant.

4.2.1 Voice Data

The experimental stimuli are pairs of six phrases derived from Pieces A and B with an overshoot that is significantly different between human groove and non-groove songs. A pair of singing data for each phrase was set up as follows:
[x] non-groove human singing
[y] overshoot-amplified singing— an overshoot of [x] was amplified to the same degree as that in the human groove song.

	x<y	x>=y	Total
Expected number	36	72	108
Observed number	55	53	108

Table 2. Comparison between amplified pitch-overshoot and loudness: (x) non-groove human singing and (y) overshoot-amplified singing

4.2.2 Procedure

The above voices were randomly shuffled [x-y, y-x, x-x]. Six students in their early twenties compared the loudness of all 18 patterns in the pairs of songs over 6 phrases. They then reported which was louder (i.e., the first, the second, or both were the same).

4.2.3 Results

Table 2 shows the results. From 108 answers, we obtained 55 answers that support our hypothesis. x<y means that the overshoot-amplified note was perceived as louder than the non-groove note. We expected 36 answers to be consistent with the hypothesis. The χ^2 test results are significant at the 5% level. This result suggests that pitch-overshoot controls perceptions of loudness. Moreover, the x>y answers only account for 10% of the x>=y answers.

4.3 Exp. 2: Comparison between Amplified Pitch Overshoot and Groove Sensation

The second experiment was conducted to verify whether pitch overshoot consonants increase groove features of a phrase.

4.3.1 Voice Data

As in the previous experiment, pairs of songs with the six phrases in Pieces A and B were prepared.
[x] non-groove human singing
[y] overshoot-amplified singing— ALL overshoots in the phrase were amplified to the same degree as those of the human groove songs.

4.3.2 Procedure

Then, 10 listeners used a web interface for the experiment. Participants listened to the twelve phrases at random through headphones or a loud speaker system in a quiet room, and they then reported degrees of groove sensation based on a 10-step system (0-9). They then described their judgment criteria.

4.3.3 Results

Figure 7 shows the listening experiment results. The horizontal axis represents the phrase number, and the vertical axis represents the average of the degree of the groove evaluation. For all of the phrases, overshoot-amplified songs presented a higher degree of groove sensation than human non-groove songs. Differences between phrases except for those related to Phrase No. 5 were significant at 5% according to the t test.

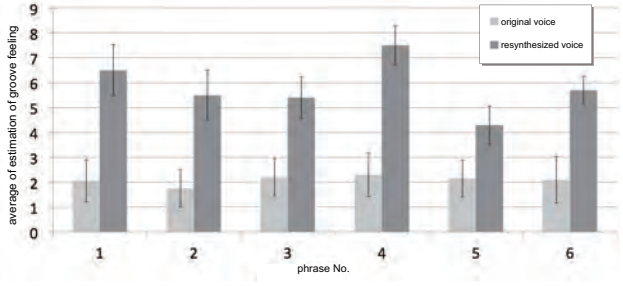


Figure 7. Amplified pitch overshoot and groove sensation listening results

This result shows that pitch overshoot control affects groove perception.

4.4 Discussion

As shown in Table 2, pitch overshoot amplification in non-groove songs made listeners experience these songs as louder. In particular, only 10% of the respondents experienced overshoot amplified songs as softer in volume. This finding suggests that there is a positive correlation between pitch overshoot and loudness perception. Moreover, one listener commented that he felt that the singer of the overshoot amplified song sang with a more pronounced accent and at a faster speed. This result supports our hypothesis that loudness is sensed more in overshoot songs even when volumes between songs are identical in practice.

As shown in Figure 7, overshoot amplified phrases were ascribed higher degrees of groove sensation than human non-groove phrases, and the five phrases were significantly different. This result suggests that amplifying pitch overshoot in a voiced consonant induces groove sensation. Several participants also commented on loudness as the listening criteria (e.g., accentuation of the start of a word). Such comments show that the perceived loudness of voiced consonants through pitch-overshoot control can either induce "groove" elements in vowels or not.

5. CONCLUDING REMARKS

As described in Section 3, in regards to the onset timing of vowels on beat, there are no differences between songs with and without "groove." The main difference between songs with and without "groove" is the length of preceding consonants and especially of voiced consonants such as "m" or "n". A lengthened voiced consonant causes listeners to sense feel a more powerful "groove." Our interpretation of this result is that pronouncing vowels on beat is indispensable for expressing accurate rhythm and that auditory illusions yielded through the control of consonants are used to express "groove." The experiment described in Section 4 was conducted to confirm this hypothesis. The results show that pitch overshoot in addition to the length of voiced consonants preceding vowels causes listeners to feel a more powerful "groove" elements. We also found that pitch overshoot of a voiced consonant preceding a vowel causes listeners to feel that a vowel is being sung louder. These results suggest that variations in

the perceived loudness of vowels stimulate "groove" sensation.

To summarize the above results, the onset timing and intensity of vowels are not essential for expressing "groove." Accuracy is given priority as a means of expressing fundamental vocal skill. Voiced consonant lengthening and pitch-overshoot are adopted to create an auditory illusion of an accentuated vowel following a voiced consonant.

Our findings reflect experiments that were conducted based on Japanese pop. Our findings may thus only apply to J-pop, which is characterized by lyrics written in mora languages. However, we believe the findings can be generalized, as they were interpreted based on perceptions of auditory illusions caused by a combination of a consonant and a vowel forming a phoneme, which are not tied to any particular language. We would like to conduct more experiments based on Korean pop, which is written in another mora language, and on American pop to identify the secrets of "groove" in terms of vocal melody.

Acknowledgments

We are grateful to Dr. Ryuichi Nariyama and Dr. Shuichi Matsumoto of Yamaha for their assistance in this study. This study was partially funded through a Grant-in-Aid for Scientific Research (C) [15K02126].

6. REFERENCES

[1] G. Madison, "Experiencing groove induced by music: Consistency and phenomenology," *Music Perception*, vol. 24, no. 2, pp. 201–208, 2006.

[2] P. Janata, S. T. Tomic, and J. M. Haberman, "Sensorimotor coupling in music and the psychology of the groove," *Journal of Experimental Psychology: General*, vol. 141, no. 1, pp. 54–75, Feb. 2012.

[3] K. Okudaira, K. Hirata, and H. Katayose, "Relationship between 'groove feeling' and the timing and loudness of drum attacks in popular music," *IPSJ SIG Technical Report*, vol. 2005-MUS-59, pp. 27–32, 2004.

[4] K. Okudaira, K. Hirata, and H. Katayose, "Relationship between 'groove feeling' and the timing and loudness of drum attacks in popular music (3rd report): Fundamental analysis of drum performance data and implementation of drum performance rendering system," *IPSJ SIG Technical Report*, vol. 2006-MUS-64, pp. 53–58, 2006.

[5] M. Wright and E. Beradahl, "Towards machine learning of expressive microtiming in brazillian drumming," in *Proceedings of International Compute Music Conference*, 2006.

[6] G. Madison, F. Gouyon, F. Ullèn, and K. Hörnström, "Modeling the tendency for music to induce movement in humans: first correlations with low-level audio descriptors across music genres," *Journal of Experimental Psychology: Human Perception and Performance*, vol. 37, no. 5, pp. 1578–1594, Oct. 2011.

[7] J. Fruhauf, R. Kopiez, and F. Platz, "Music on the timing grid: The influence of microtiming on the perceived grove quality of a simple drum pattern performance," *Musicae Scientiae*, vol. 17, no. 2, pp. 246–2690, 2013.

[8] G. Madison and G. Sioros, "What musicians do to induce the sensation of groove in simple and complex melodies, and how listeners perceive it," *Frontiers in Psychology*, vol. 5, no. 894, Aug. 2014. [Online]. Available: http://dx.doi.org/10.3389/fpsyg.2014.00894

[9] G. Sioros, M. Miron, M. Davies, F. Gouyon, and G. Madison, "Syncopation creates the sensation of groove in synthesized music examples," *Frontiers in psychology*, vol. 5, Sep. 2014. [Online]. Available: http://dx.doi.org/10.3389/fpsyg.2014.01036

[10] M. Goto, "Singing information processing," in *Proceedings of the 12th IEEE International Conference on Signal Processing (IEEE ICSP 2014)*, October 2014, pp. 2431–2438.

[11] T. Nakano, M. Goto, and Y. Hiraga, "An automatic singing skill evaluation method for unknown melodies," *Journal of Information Processing Society of Japan*, vol. 48, no. 1, pp. 227–236, Jan. 2007.

[12] T. Nakano and M. Goto, "Vocalistener2: A singing synthesis system able to mimic a user's singing in terms of voice timbre changes as well as pitch and dynamics," in *Proceedings of the 2011 IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP 2011)*, May 2011, pp. 453–456.

[13] H. Kawahara, T. Irino, and M. Morise, "An interference-free representation of instantaneous frequency of periodic signals and its application to f0 extraction," in *Proc. ICASSP 2011*, May 2011, pp. 5420–5423.

[14] M. Morise, "An attempt to develop a singing synthesizer by collaborative creation," in *Proc. the Stockholm Music Acoustics Conference 2013 (SMAC2013)*, Stockholm, 2013, pp. 287–292.

[15] T. Saitou, N. Tsuji, M. Unoki, and M. Akagi, "Analysis of acoustic features affecting "singing-ness" and its application to singing-voice synthesis from speaking-voice," in *Proc. Inter Speech - ICSLP*, 2004.

[16] E. Narmour, *The analysis and cognition of melodic complexity: the implication-realization model*. University of Chicago Press, 1992.

[17] Y. Ohishi, H. Kameoka, D. Mochihashi, and K. Kashino, "A stochastic model of singing voice f0 contours for characterizing expressive dynamic components," in proc. international conference on spoken language processing," in *INTERSPEECH 2012*, Sep 2012.

IMPROVISATION AND GESTURE AS FORM DETERMINANTS IN WORKS WITH ELECTRONICS

Alyssa Aska
University of Calgary
alyssa.aska@ucalgary.ca

ABSTRACT

This paper examines several examples that use electronics as form determinants in works with some degree of structured improvisation. Three works created by the author are discussed, each of which uses gestural controller input to realize an indeterminate form in some way. The application of such principles and systems to venues such as networked performance is explored. While each of these works contains an improvisatory and/or aleatoric element, much of their content is composed, which brings the role of the composer into question. The "improviser", who in these works advances the work temporally and determines the overall form, is actually taking on the more familiar role of the conductor. Therefore, these works also bring up important conversation topics regarding performance practice in works that contain electronics and how they are realized.

1. INTRODUCTION

Improvisation in some form permeates nearly all genres and eras of music: the baroque had improvised keyboard music, improvisatory solos are an integral part of jazz realization, and even classical concerti contain cadenza sections, which are improvised by a soloist. When one considers the ornamentations that were assumed in certain styles of baroque music, as well as temporal improvisation such as rubato in romantic piano music, it is apparent that some degree of performance liberty, or interpretation, is applied to nearly all music, no matter how precisely the notation of the work. Electronic systems, particularly live ones, often have some elements that change, even if slightly, based on performance conditions. This is true of acoustic instruments to an extent (elements such as reverb and delay are dependent on venue, and musicians may compensate for such and perform differently) as well. Taking into account computational processing lag, minute differences in algorithm realization, and live interactive components that may differ in each performance, it is clear that the use of electronics is adapted well to flexible performance realization. It is this flexibility requirement that led composers to seek live systems in favour of fixed media coupled with instruments in the past – for many works that included tape alongside live instruments, there simply wasn't enough room for the performer to give a fluid performance whilst also adhering to the strict time of the tape. (This is not the case, however, for all tape music, when one considers some built-in flexibility in works such as Mario Davidov sky's *Synchronisms, no. 6*) [1]. Electronic improvisation has also been explored, in programs such as Omax, in which the computer actually improvises along with a live instrument [2]. This paper focuses primarily on improvisational or aleatoric elements that affect the large-scale form or other temporal elements of a work. The works discussed below all make use of gestural control in some capacity to determine form and flow of time; that is, in each of these works, an performer interacts with a gestural controller to determine either the overall form of the work, or durations of sections within.

2. PRECEDENTS

While improvisation and multiple score interpretation can be observed in many periods throughout history, an exhaustive discussion of improvisatory works is beyond the scope of this paper. Therefore, the practical discussion is preceded with a brief description of some of the more relevant recent works (and scholarship) involving improvisation and/or indeterminacy. The music of John Cage presents an excellent starting point for this discussion, as many of his works contained indeterminate procedures within a modern framework. Many of these works resulted in incredibly diverse interpretations, and Philip Thomas noted that, regarding his *Solo for Piano* (1957-58), "No single interpretation is likely to represent all the material available, and the multiple possibilities open to the interpreter within just a single piece make the *Solo for Piano* a work that is uncontainable: it resists definition, and at micro- and macro- levels the score can be only the beginning of a process, a prompt for action rather than a description of sound" [3]. Thomas further commented on several interpretations of this work, describing that the score was not a definitive element in much of Cage's work, and that the resulting music was substantially diverse from interpretation to interpretation. When electronics become a consideration, the possibilities for improvisation or indeterminacy increase, especially for those works that involve computing. However, most musicians are not trained to improvise with a computer system (and while growing, pedagogy surrounding the performance of works for instrument and live electronics remains limited). Chapman Welch, who composed several works involving improvisation with an electronic entity, remarked that "the addition of an interactive computer system within an improvisation adds a

new layer of musical input and complexity that is foreign to many improvisers" [4]. Welch describes some of the processes he uses to make improvisation with electronics more approachable for musicians through his work on the piece *Moiré,* for clarinet soloist, computer, and ensemble. The result is highly controlled improvisational blocks, as well as a conductor that gives cues. Welch describes further his future work in storing the musicians' real time choices in a database and using such data to affect the performance (thereby creating more of a two-way improvisation and effect). This type of two-way approach was also implemented in software designed at IRCAM titled Omax. The software, which combines the software Open Music with Max, "learns in real-time typical features of a musician's style and plays along with him interactively, giving the flavor of a machine co-improvisation" [5]. The creators of Omax describe it as an "improvisation-oriented musician-machine interaction system that learns in real time from human performers" [6]. Omax has been implemented successfully in numerous performances, and indicates that substantial research in the designing of such systems has been completed, and that improvisation with electronics represents a viable performance genre. Both of the systems discussed here represent effective means of working with improvisation and indeterminate elements specifically in working with a computer. They take a different trajectory however; electronics are incorporated as a primary improvisatory driver, but the nature of the improvisation resembles that of the Cage example discussed above, or another Open Form work. However, some elements, such as Welch's use of conductor and improvisational feedback loop, do find analogues in elements of my works. The other feature of the works described in this paper is the use of gesture and motion capture systems as an integral component of the improvisation.

3. EARLY EXPLORATION – *THE BLACK SWAN*

I created a work, performed twice in late 2013 and presented as a poster at the 2014 ICMC-SMC joint conference, entitled *The Black Swan,* for performance over a network [7]. *The Black Swan* consisted of a graphical score, which was to be realized during performance, a score for "movers", who to some extent generated the score, and a master score, which transferred performance instructions. All of these scores consist of software components created using Max, and made use of Jacktrip to facilitate the audio transferal over the network [8, 9]. One performance node was tracked with a webcam, with captured movements translated into data using computer vision algorithms developed for Max [10, 11]. While the presence of a score indicates that there is some "composed" or "fixed" element of the piece, the actual form is somewhat improvisatory, as the flow of time through the score is generated during performance. This is executed the following way: one node receives software containing the performance score, which the performer (or performers) reads off of the computer screen in real time. This performance score is intended to be read by trained musicians, but is in open score format, allowing any ensemble to perform the work. Therefore, the score contains graphical elements, but the basic layout of the score is very familiar to anyone that understands western concert music notation: there is a staff and symbols placed upon it are read left to right, with vertical dimension indicating relative pitch, which is dependent upon the performing instrument's range. Another node contains instructions for a performer to interact with a motion tracking system; this interaction serves to make small modifications in the advancement of the score, as well as, at pivotal moments, major score changes.

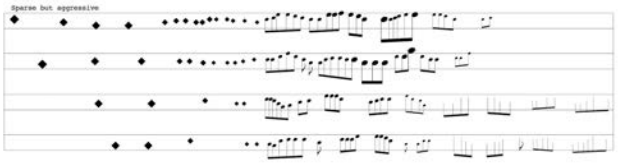


Figure 1 The Black Swan score image

This system allows for the kind of structured improvisation that integrates well with electronic interactivity and networked performance, as the latency involved in network systems changes depending on how far away the two performance nodes are, and what type of network connections they have. This makes precise rhythmic interactions over networks potentially difficult and even problematic [12]. *The Black Swan* allows two performance nodes to be geographically separated and still maintain an engaging performance without performance being compromised by the latency of the network. The motion-tracking node receives instructions for performance from a master controller (a role that in the past has been performed by myself). The master sends text information in real time over the network to the performers on the motion tracking side regarding what type of movements to execute, which direction to move, how much of the body to move, etc. This allows for a significant amount of indeterminacy from performance to performance, as the number of performers on the motion tracking side can vary.

Figure 2 The Black Swan camera tracking software

These performers have a visual display, which indicates a number between one and one hundred, and when the

number reaches one hundred, the score advances to the next section. The tracking used for this is very general, with the computer vision algorithms evaluating the overall direction and amount of movement. The real-time text-based performance instruction creates the performance interest, because the master controller can instruct the movers to do anything. The movement also causes adjustment of the observable score image at any given time. Since the musical performers are instructed very specifically to read the score left to right, this changes the structure of the piece as well. Further, at some point in the score, the motion tracking team is actually instructed to perform some musical sound, which is sent to the score realization team, unbeknownst to them. This instruction is given by the master, and also adds an element of indeterminacy and surprise to the work.

4. *THE WOMAN AND THE LYRE* PART I*: FAYUM FRAGMENTS*

Fayum Fragments is an individual poly-work that serves as part of a larger poly-work, entitled *The Woman and the Lyre,* for mezzo-soprano, flute, cello, piano, and live electronics. In this paper, the work *Fayum Fragments* will be discussed as its own individual entity. Specific attention will be paid to the form of *Fayum Fragments,* and how the use of gestural controller is integral to the realization of the overall form of the work. The text that serves as inspiration and setting of the work consists of a series of short Greek fragments extracted from a surviving poem by the Archaic Greek woman Sappho, titled the *Fayum Fragments* [13]. The poem, the second of two appearing in the *Fayum Fragments,* consists of ten lines, of which eight have discernible text. Ten musical miniatures were composed as a response to this poem; each of the miniatures represents the only discernible word or phrase in its line. This actually provided a large amount of inspiration regarding the form of the piece; the meaning behind the poem cannot be determined without the context of the words, and this work seeks to create its own meaning every time it is performed by framing the "words" in a different context. While each of the short instrumental movements can function as a standalone musical work, they are designed for performance either simultaneously or successively (or a mixture of both) as part of larger sections or movements, of which there are three.

4.1 *Fayum Fragments* – form and text

The overall performance structure of the *Fayum Fragments* is as follows:

1) Fragments 1 – Performance of two or three fragments, concluded by a short tutti fragment
2) Fragments 2 – Performance of two or three fragments, concluded by a short tutti fragment
3) Fragments 3 – Performance of two or three fragments, concluded with silence/end

As discussed previously, each singular fragment is inspired by the sound of a small fragment in Greek derived from a larger, and mostly unreadable, ancient poem. The phrases have been translated into English by Henry T. Wharton to mean: 1) *soul*, 2) *altogether*, 3) *I should be able*, 4) *as long as indeed is to me*, 5) *to flash back*, 6) *fair face*, 7) *stained over*, and 8) *friend*. There were two lines in the original poem of which no text was discernible, and the two tutti (instrumental only) movements in the work represent these by incorporating the (interpreted) rhythmic structure of the lines.

4.2 Aleatoric elements

Fayum Fragments contains ten pre-composed, short musical movements for live performance. However, the order in which they are performed, and the time passing between onsets of each fragment, is determined entirely by the vocalist during the live performance. The vocalist uses a Leap Motion controller, reading off of a gestural score, which instructs her to execute specific hand motions. *Event detection* is used to trigger the onset of new movements once the vocalist performs pre-determined actions. Each time the vocalist triggers a new fragment, the system "looks" for a different circumstance. For example, at one point the software will allow a fragment to be triggered as soon as the vocalist places both hands within the tracking area.

Once this fragment is triggered, however, the software will move on to detect another gestural event. An algorithm is used to determine which movement begins when a new one is triggered. In the very beginning of the *Fayum Fragments,* any of the fragments may be triggered. However, once the work begins, each section only allows for fragments that don't contain instruments that have already performed or are currently performing to be triggered for performance. Since each fragment consists of a different subset of the instrumental group (or a solo instrument), this allows for the fragments to be ordered somewhat randomly.

4.3 The use of the Leap Motion to generate form

The vocalist uses a Leap Motion controller to trigger the onset of fragments, and thus determine the form of *Fayum Fragments* during performance. The Leap Motion interfaces with Max using the MRLeap object [14], and tracks the singer's hand movements along X, Y, and Z-axes, as well as velocity measurements. A gestural score is given to display which gestures the vocalist is to make throughout the performance of each fragment (or fragments). This score is very detailed regarding the type of gestures, their size, which hand to use, and the location on a given axis to start or finish them. However, the vocalist is instructed to perform each section at any speed desired, and is given instructions to leave as much (or as little) time as desired between the onset of fragments. As the order of the fragments is somewhat random, this enables the vocalist to make a decision during the perfor-

mance regarding how quickly to progress through each section. Therefore, the duration of *Fayum Fragments* is highly variable, and can last anywhere from about ten minutes in duration to about twenty minutes in duration.

The work uses a *score-oriented event detection* procedure, since the Max software listens for specific gestures to occur during given points to trigger the performance of each new fragment [15]. Originally, I had considered using specific gestures to control the advancement, or triggering of fragments.

This procedure would have been incredibly simple, as the Leap Motion already has traceable built in gestures, which could have served as triggering elements. However, this would have removed a dimension from the composition that is extremely important, which is that of the singer's role of conductor, or form-determinant, of the work. Since the work contains many other dramatic elements, this was particularly important, and incorporating the flexible and notated gestures allows for the interaction between the singer and the Leap Motion to contain its own drama and narrative that threads through the entirety of the work. This in effect gives a distinct and meaningful form to a work that has so many elements of indeterminacy.

5. *OPEN SPACE:* EXPANDING *THE BLACK SWAN*

A follow-up work to *The Black Swan* was created in 2015-2016, entitled *Open Space.* This work is intended for performance between two geographically displaced nodes, although it can also be executed between two parties onstage. The work uses motion tracking as a means for score advancement, with detailed instructions a performer must follow, but a loose temporal structure. *Open Space*, however, contains more direct interaction between the two nodes, involving motion tracking in both locations that results in sound and formal advancement.

5.1 Space modification node

The space modification node consists of a motion tracking system, which the performer is to engage with. The performer is given loose instructions to explore the physical space of the room or performance area. This exploration serves two purposes: 1) it selects the sound files that will play back and be sent to the sound modification node to perform with, and 2) to spatialize the audio generated by the sound-generating performer. The sound file selection is relatively simple: the video feed is divided into four separate tracking areas. Each of these areas corresponds to one of the environmental soundfiles, and whichever area has the most perceptible motion at any given moment determines which sound file is played back. A crossfade is implemented as well, so that when the performer moves from location to location, the change between sound files is smooth. The location of the performer also results in minor modifications of the environmental sounds, such as delay times (which get longer as the performer moves further from the centre points), delay feedback (which increases as the performer moves closer to the centre points) and other modifications which simulate distance in some way, such as low pass filter parameters, and output volume. At some point in the piece, the performer will "discover" a location with a bell sound, a discovery, which serves to determine the duration of the piece. Once the bell sound is triggered, the sound modifier is given a very fixed performance score to execute, with instructions that the piece will end at its' completion. The space modifier is also instructed to remain in the space he/she is standing and to only use arm movement to diffuse sound at this time.

5.2 Sound modification node

The sound modification in *Open Space* occurs as a result of a performer at a second node also interacting with a camera. The sound-modifying performer is given instructions within the performance materials regarding which gestures, actions, and movements to perform, and a graphical score that they are to execute during a certain moment in the performance. The characteristic of the sound that is generated depends on where, how fast, and how large of a motion a performer makes. However, the exact sounds that are triggered are largely depended on the *space modifier*. Unlike *The Black Swan,* in which the score generator actually advanced the score and determined the form temporally, the space modifier in *Open Space* interacts with a camera to provide the sound generator with an environmental sound file. All of the sounds are derived from recordings taken from various locations, both indoor and outdoor. The sound modifier "explores" each space, as the sound material from the score generator is recorded to a storage buffer that is sent to the sound modifier over the network. The sound modifier then uses gesture and movement to "play" this stored sound as an instrument, using a granular synthesis, and adjusting various parameters of the grains. The computer vision used in *Open Space,* therefore, while making use of the same software as *The Black Swan*, uses the data in much more specific ways, with direct links between movement and resultant sound.

5.3 Bell sequence

At an unspecified point during *Open Space,* the space generator will choose to "discover" a space no longer occupied by a location, but by bell sounds. This action causes the work to end; the space modifier can no longer change the environmental sound file. Once the bell sequence is triggered, the space generator serves as sound diffuser, using the motion tracking to move the sounds around the space. The sound generator follows a gestural score during this sequence, and upon completion, the piece is finished.

Open Space is therefore, while being quite different in form from *The Black Swan* and *Fayum Fragments,* quite similar in that the overall form and temporal pacing is moderately indeterminate. Although some ele-

ments of *Open Space,* such as the fixed score at the end, are very strictly composed, the overall form of the piece is not. Again, this type of performance is well suited to networked performance environments, in which there may be lagging of both audio and video environments.

6. FUTURE DIRCTIONS: *SAPPHIC CYCLE*

Thus far I have explored those elements of electronic music that pertain mostly to the electronics, and specifically, to gestural control. In all the previously discussed works, gesture control to some effect determines the overall form and duration of the piece. I will now explore some of the compositional (and as an extension, notational) elements I am currently developing that are effective at handling some of the issues that arise when dealing with live electronics in combination with acoustic instruments that have a somewhat fixed score, such as unknown length of effect, the desire for interactivity, and others.

6.1 Previous explorations – Redshift

My first experience with a type of aleatoric notation began when I was commissioned to contribute a composition for a spatialized ensemble [16]. This ensemble, which titled itself a *Vertical Orchestra*, performed in various venues, and each time they performed they were spread out throughout the space, rather than close together as a traditional ensemble. This presented some challenges for composition, most notably the lack of visibility of a conductor and other ensemble members. This made the execution of precise, synchronized rhythms difficult, and favoured more flexible notation. To adapt to this medium, I developed a score that was primarily structured improvisation, using boxes to determine the parameters to be performed, and lines and durations to determine how frequently to perform the actions, and whether sound should be continuous or have silence.

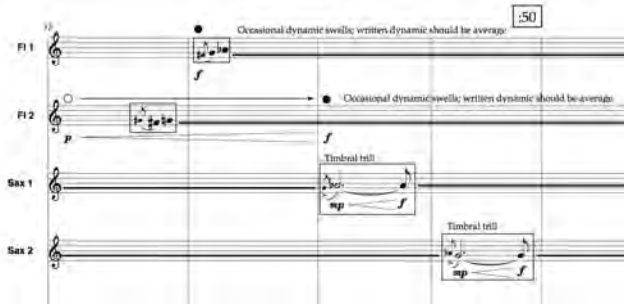


Figure 3 Score image from Redshift composition

Many of the issues involved here are also present in networked music (such as lag time, and lack of complete visibility), which contributed to the reason behind my desire to use a more open, aleatoric-inspired notational system for my networked pieces. However, this type of notational thinking also lends itself well to performances in which live instruments interact or perform alongside live electronic systems. It was for this reason that I decided to develop this type of notation further in *Sapphic Cycle.*

6.2 *Sapphic Cycle* – notational elements

Sapphic Cycle represents the other part of the larger polywork *The Woman and The Lyre.* The cycle consists of four songs, whose texts are inspired by fragments of Sappho's texts. These texts are all in English, written in the early twentieth century by Bliss Carman [17]. *Sapphic Cycle* also incorporates motion tracking of the instruments, however, rather than using the Leap Motion, the instrumentalists and the vocalist are tracked via cameras. Creating a flexible notation and interaction system for these songs presented a challenge; I wanted to retain some sense of rhythm, and precise pitch, while also allowing the performers to interact with the electronics in a meaningful way. Therefore, the majority of the cycle is composed in a way that is very close to standard western notation. To allow for flexibility and interactivity with the electronics, I have chosen two means: 1) larger areas with precise rhythmic motives but indeterminate "pausing" time, to allow for the music to breath and the electronics to respond, and 2) areas or blocks in which the performers can "break out" of the metered performance and improvise within the notated parameters. Both of these types of notation allow primarily for temporal flexibility; the pitches, harmonic structures, and rhythmic gestures remain intact. I also have created a number of symbols, which indicate more general parameters; this was done to retain the concept of gesture that permeates the work, and also to provide visuals that are not obtrusive but can convey a multi-dimensional meaning. An example of this is a symbol that is used to indicate that the performers, when within the box, are to perform the notes from any starting point and going in any direction. Words are also used symbolically, to project intentions, emotions, or overall musical concepts. These words can be expressive (such as "lyrically", "harshly", "agitated", etc.) or more action-oriented (such as "as fast as possible"). To separate these intention words from general musical terms used in the score, these words contain a boxed frame.

6.3 *Sapphic Cycle* –implications and interactivity

The two networked pieces described above (*The Black Swan* and *Open* Space) both provide an interesting performance environment for networked concerts and allow the performers to have a large amount of control over the resulting work. However, this type of temporal indeterminacy is not suited for works that require finer control over elements such as pitch, dramatic action, and precise counterpoint. *Fayum Fragments* allows greater control on part of the composer as the scores themselves are fixed, but since the arrangements are indeterminate, this work does not achieve the type of precision a composer may desire in a traditional chamber work. The flexibility in *Sapphic Cycle* is not as dramatic, but at the same time allows the performers to interact with the electronics in a way that they feel is meaningful. The work does not have a diverging or reversible timeline in the way that the previous three works do, yet moving forward, time has some flexibility. As so much of the electronic processing, including spatialization and delay times/feedback, is deter-

mined by the performers' specific gestural actions during performance due to the motion tracking, instructing the performers as to when and how they can modify their own playing gives them greater control over their response and interaction with the electronic sound.

7. CONCLUSIONS

Music involving live electronics lends itself to improvisation because of the very changeable nature of electronics, and the ease of implementing indeterminate processes. Several systems have been developed that allow either an acoustic instrumentalist to improvise over electronics, or electronics to improvise in some way with an acoustic instrument. The systems described in this paper serve a very different function than those developments, as the improvisatory and indeterminate elements are very linked to the overall form of works, and especially to the drama and program. These systems integrate gesture for various reasons; such as creating an organic link between musical gesture and electronics (as both are technically effected by the same physical movement at times, such as in *Sapphic Cycle)*, or acting as a type of conductor, such as in *Fayum Fragments.* Conducting has its own visual gestural component to it, which can be highly dramatic – *Fayum Fragments* aims to exploit this component. Finally, in networked systems (*The Black Swan* and *Open Space*) gesture is desirable because it allows both an audible and visual link between nodes, thus enhancing the feedback between groups that are not physically close, and increases the potential performer pool. Highly skilled musicians as well as non-musicians explorers in a setting such as an installation can perform both of these networked pieces.

8. REFERENCES

[1] M. Davidovsky, *Synchonisms no. 6.* E.B. Marks Music Corp., 1972.

[2] G. Assayag, G. Bloch, A. Cont, and S. Dubnov, "Interaction with Machine Improvisation," in *The Structure of Style,* Berlin-Heidelberg, Germany: Springer, 2010.

[3] P. Thomas, "Understanding Indeterminate Music through Performance: Cage's Solo for Piano," in *Twentieth-Century Music*, vol. 10.01, pp. 91-113, 2013.

[4] C. Welsh, "Programming Machines and People: Techniques for live improvisation with electronics," in *Leonardo Music Journal,* vol. 20, pp. 25-28, 2010.

[5] "The Omax Project Page." <http://omax.ircam.fr/>

[6] G. Assayag, et al. "Omax brothers: a dynamic topology of agents for improvization learning," in *Proceedings of the 1st ACM workshop on Audio and music computing multimedia*. ACM, 2006.

[7] A. Aska, "*The Black Swan:* Probable and Improbably Communication Over Local and Geographically Displaced Networked Connections as a Musical Performance System," in *Proceedings of the International Computer Music Conference-Sound and Music Computing Conference,* Athens, Greece, 2014, pp. 553-556.

[8] Zicarelli, D. Max/MSP Software. San Francisco: Cycling '74. 1997.

[9] J. Cáceres and C. Chafe, "JackTrip: Under the hood of an engine for network audio," in *Journal of New Music Research,* vol 39.3, pp. 183-187, 2010.

[10] Pelletier, J. "cv.jit – "Computer Vision for Jitter." <http://jmpelletier.com/cvjit/>

[11] February 15, 2012. Ritter, M., "MR.jit.toolbox", < http://www.martin-ritter.com/software/maxmsp/mr-jit-toolbox/>

[12] Chafe, Chris, et al. "Effect of time delay on ensemble accuracy." *Proceedings of the International Symposium on Musical Acoustics*. Vol. 31. 2004.

[13] H. T. Wharton, *Sappho: Memior, Text, Selected Renderings, and a Literal Translation.* Amsterdam, Netherlands: Liberac, 1974.

[14] M. Ritter and A. Aska, "Leap Motion As Expressive Gestural Interface," in *Proceedings of the International Computer Music Conference-Sound and Music Computing Conference,* Athens, Greece, 2014, pp. 659-662.

[15] S. Emmerson, *Living Electronic Music.* Burlington, VT: Ashgate Publishing Ltd., 2007.

[16] http://www.timescolonist.com/entertainment/music/audience-at-centre-of-vertical-orchestra-1.572424

[17] B. Carman, *Sappho: One Hundred Lyrics.* A. Moring, The De La More Press, 1906.

FORM-AWARE, REAL-TIME ADAPTIVE MUSIC GENERATION FOR INTERACTIVE EXPERIENCES

Christodoulos Aspromallis
Department of Computer Science
University College London
c.aspromallis@cs.ucl.ac.uk

Nicolas E. Gold
Department of Computer Science &
UCL Centre for Digital Humanities
University College London
n.gold@ucl.ac.uk

ABSTRACT

Many experiences offered to the public through interactive theatre, theme parks, video games, and virtual environments use music to complement the participants' activity. There is a range of approaches to this, from straightforward playback of 'stings', to looped phrases, to on-the-fly note generation. Within the latter, traditional genres and forms are often not represented, with the music instead being typically loose in form and structure. We present work in progress on a new method for real-time music generation that can preserve traditional musical genres whilst being reactive in form to the activities of participants. The results of simulating participant trajectories and the effect this has on the music generation algorithms are presented, showing that the approach can successfully handle variable length forms whilst remaining substantially within the given musical style.

1. INTRODUCTION

A significant portion of current artistic and entertainment narrative-based experiences are structured episodically and the duration of these episodes (i.e. sections) may be only loosely determined due to semi-improvised content or participant-dependent advancement of their narrative. Such experiences include live theatre containing improvisation [1], theme parks [2, 3], extended theatrical interactive experiences [4], video games, interactive film [5] and virtual environments. We hereafter refer to all these as Extended Performance Experiences (EPE).

Musical accompaniment is well established in EPEs due to its potential for a more convincing and engaging experience on a "cultural, physical, social or historical level" [6]. In addition, music can play a narrative-defining role and, thus, significantly enhance the experience [7].

Computer generation of musical accompaniment for EPEs is necessary for a number of reasons: The physical presence of musicians may be counter-immersive or impractical e.g. where, owing to limited stamina of humans, rotation may be required during a long-running experience. This could lead to musical incoherence due to subjectivity of musical decisions if musical improvisation is involved. In addition, the resources required for rehearsals and performances may not be economically achievable for small-scale projects or independent companies.

It is common that EPE episodes are contrasting in content and aesthetic character and, thus, musical accompaniment needs to reflect this (for example, one might imagine four distinct zones within a single theatrical installation, each with a different cultural flavour and thus musical style). The transitions between episodes are an important aspect of the overall experience since, according to Benford *et al.* [8], transitions are benchmarks in continuous interactive experiences in space and time. Musical transitions (MT) must take place at appropriate narrative boundaries and without breaking musical continuity.

The indeterminate duration of EPE episodes, combined with the necessity of timely and well-placed musical transitions, raises the issue of musical coherence and continuity, i.e. we wish to avoid abrupt MTs or unexpected silences. In order to avoid such abruptness during an EPE transition, various techniques have been developed for real-time generation of music (see section 2).

We can summarise the requirements for a music generation system for EPE as: the need to provide continuous, non-repetitive music, with non-abrupt but distinct musical transitions, with large-scale form awareness and with note/chord level granularity. By large-scale form we mean musical forms that might be traditionally recognised e.g. blues, pop songs, binary, tertiary etc.

This paper addresses the problem of generating well-formed music incorporating timely transitions in the presence of indeterminate and dynamically changing compositional length in real time. We present the Form-Aware Transition Engine (FATE): an approach that combines probabilistic music generation with participant trajectory estimation to permit changes to be made to musical structure in real time within the constraints of a style (represented by a probabilistic grammar).

The rest of the paper presents background (Section 2), the FATE approach (Section 3), case studies (Section 4), results (Section 5) and conclusions (Section 6).

2. BACKGROUND

There has been a substantial amount of work on generative music (see [9] for a survey) and in particular real-time [e.g. 10, 11, 12, 13] and non-real-time [14, 15] generative music in EPE-like settings. The real-time work is of particular relevance here and has resulted in a number of common approaches (see summary in Table 1).

2.1 Pre-composed passages played once

This approach is traditionally taken in theatrical performances, in the principle of television 'stingers' [16], by manually triggering or conducting musical passages of shorter duration than the narrative episode lasts. As a result, this approach does not provide continuous musical accompaniment of the narrative.

2.2 Pre-composed consecutive or looped passages

This approach is common throughout the history of music for video games (Super Mario Series – 1985 onwards, Halo – 2001, Earth Eternal – 2009, among others). Sudden stoppages and starts of music passages in early video games were later replaced by volume crossfading [6]. Despite this, players have characterised such MTs as abrupt and loops as monotonous [6]. Müller and Drieger [10] developed a system for automated, real-time manipulation of pre-existing musical clips in response to narrative action in visual media. The system manages time-placement and concatenation of precomposed music. Hazzard [17] developed an adaptive music soundtrack for a musically augmented walking experience in Yorkshire Sculpture Park. The dynamically managed composition relies on triggering and looping short segments of music or the alignment of longer ones.

The lack of note-level flexibility in precomposed music may lead to abrupt changes in music and general inconsistency between music excerpts thus rendering it unsuitable for the problem being addressed here.

2.3 Dynamic management of pre-composed layers

In this approach, predefined melodies, rhythmic patterns, chord progressions or baselines that are musically compatible are vertically managed, i.e. triggered and stopped. Early instances of this approach include Langston's 'riffology' [18] for the Atari console game *Ballblazer* [19], an engine that linearly connects precomposed melodies based on interval connectivity as well as Whitmore's instrumental layer approach [20] for Microsoft's *Russian Squares* [21]. In the more recent music engine prototype [.talktome] [22] and in the recently released video game *NoMan's Sky* by Hello Games [11] music layers are also algorithmically managed, based on game state changes.

Dynamic management of precomposed layers provides arrangement flexibility and, specifically, the ability to punctuate MTs through the use of different instrument settings between music sections. However, the granularity of predefined music layers may not correspond to the granularity of the narrative and so abruptness is still a risk. This may also be musically restraining.

2.4 Note-level procedural generation of original material

Video games like *Journey* [12], *Spore* (music by Brian Eno) [23] and *Simcell* [24] generate an ambient music floor as well as melodic and rhythmic motifs in response to game states and events. However, the non-metric and ambient character of the music does not generate distinctively contrasting music sections. Even though musical accompaniment in these instances meets the needs of the specific video games, other EPEs may require musical contrast at the level of harmonic context, rhythm and large-scale musical form, which is something that these instances of note-level generation do not provide.

2.5 Note-level morphing

Beyond the aforementioned EPEs, Wooller [25] applied morphing programming techniques to music loops in order to generate MTs between mainstream pieces of popular dance music. This approach sets an initial loop as a starting point and a second as a goal point. Systems developed by Wooller [25] and Brown, Wooller and Thomas [13] can generate MTs in the form of music morphs between loops with music generation decisions at the note-level. The *Morph Table* [13] can control the progress of these morphs in real time using interfaces such as moving cubes on a table. Morphing was achieved by using parametric, probabilistic and evolutionary techniques. However, the system manipulates short music excerpts (i.e. loops) and does not take large-scale form into account. Instances of EPE episode transitions may require music to either develop formally or to simply stop by abiding to specific form rules. For instance, on-demand ending of a blues form should be managed in a way that preserves its character and does not sound imbalanced, as might be the case if it ends at an arbitrary point.

3. FORM-AWARE TRANSITION ENGINE

We present a Form-Aware Transition Engine (FATE) that aims to address the problem of MTs in EPEs.

3.1 Architecture

The FATE design consists of two main modules: a *prediction engine* provided with external stimuli and a *music generation engine*. The *prediction engine* receives data from an EPE environment, such as the location and movement of an EPE participant from a Microsoft Kinect™ sensor. Based on environment data, the prediction engine estimates the remaining time towards the next Narrative Transition Point (NTP): the point at which the current episode of music should finish. In the current implementation of FATE, the prediction engine receives its data from a mouse-controlled 2D surface of the computer screen (Figure 1). This simulates a simple version of an EPE episode, where a participant, represented by the dot (top left corner of Figure 1), is expected to arrive at the goal point (bottom right corner) after some time. Reaching the goal denotes the end of the episode. The prediction engine provides the music engine with data based on the three following elements:

(i) An exponentially weighted moving average (EWMA) of minimum remaining time towards an episode end. To calculate this, an EWMA of the absolute speed (irrespective of direction) of the participant is computed continuously. Based on the participant's distance from the goal

Musical elements / Generation approaches	Continuity	Non-abrupt development	Distinct Transitions	Repetition avoidance	Granular control (note/chord - level generation)	Large-scale form
Precomposed passages-once				X		
Precomposed passages-consecutive/looped	X		X			X
Precomposed layers	X		X	X		X
Note-level procedural generation	X			X	X	
Note-level morphing of loops	X		X		X	
Hypothesised algorithm	**X**	**X**	**X**	**X**	**X**	**X**

Table 1. Summary of real-time music generation approaches for EPE-like settings.

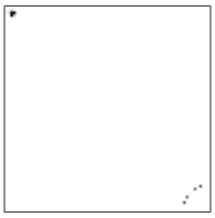


Figure 1. Mouse-controlled 2D surface.
Dot / participant (top-left), goal point (bottom-right).

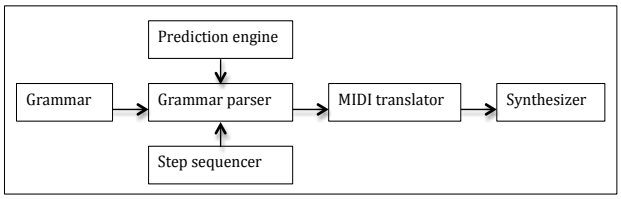


Figure 2. FATE design.

point at a given time, the remaining time is computed under the assumption that a direct linear path towards the goal is taken at that time.

(ii) Binary information on whether the participant is approaching or moving away from the goal point.

(iii) Binary information on whether or not the participant has reached the goal.

The signals sent from the prediction engine to the music engine are configured thus: If (i) is less than a specified duration appropriate to the musical form being generated (e.g. the duration of four bars in the following case studies – see Section 4) and (ii) the participant is approaching the goal, then a signal ("ENDING") is sent to the music engine to request that a process to end the music should begin (*cadencing* - Section 3.2.3). If "ENDING" is true and either i or ii cease to apply, then the "RECOVERY" signal is sent. Finally the "GOAL_REACHED" state becomes true at the end of the episode and is irreversible. The *music engine* generates chords in real time in order to populate (currently) a blues twelve-bar form (the particular musical style required is captured by a probabilistic grammar). Data received from the prediction engine enables the music engine to manipulate the generation of chords so that the musical end matches the episode end in a timely fashion, i.e. reaching the goal.

MTs must happen in a way that makes musical sense in the context of both the finishing and upcoming music sections. At this stage FATE has been developed to tackle the first part of this issue i.e. ending and recovering (more details in Sections 3.2.3 to 3.2.6) a finishing music section in such a way that musical form is retained or at least stopped in as least an abrupt manner as possible. At this stage we use the twelve-bar blues because it is a well-defined form of medium length.

Figure 2 presents the FATE design. Namely, the grammar parser loads a set of structural attributes and grammar rules to the system. A step sequencer drives and continuously updates the system on musical time. Grammar productions happen according to musical time as well as incoming data from the prediction engine. At each step of harmonic progression – in this case defined at one chord per bar – a chord symbol is sent to the MIDI translator which finally outputs the MIDI notes (chords) to a virtual synthesizer.

3.2 Grammar

The generation of chords relies on a hand-built, stochastic, context-sensitive generative grammar. A grammatical approach was chosen as grammars are good for supporting the management of the "macrostructure" of form [26].

Music generation is driven by two main types of decisions: a) musically-driven decisions (3.2.1), defined by the norms of the style (here blues), and b) event driven decisions (3.2.3 to 3.2.6), defined by incoming data from the prediction engine.

3.2.1 Temporal Resolution of Hierarchical Decision-Making (Form-Dependent)

The top-level of chord decision-making is made at specific points of the blues structure. Specifically, *top-level*

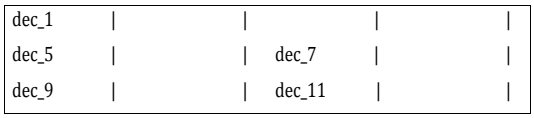


Figure 3: Top-level decision points.

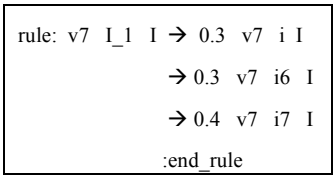


Figure 4: Context-aware time-specific rule for tonic in bar 1.

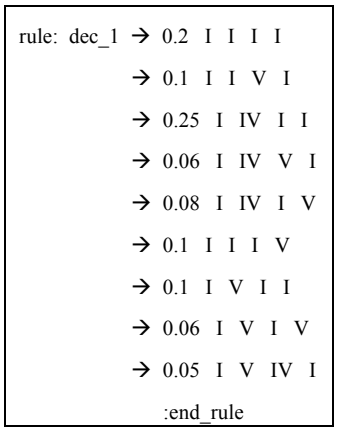


Figure 5: LHS is a time-specific (bar 1) top-level decision rule. On the RHS each production probability is defined along with the type-level non-terminals pro-

decisions are made in bars 1, 5, 7, 9 and 11 of the form as shown in Figure 3 (denoted as 'dec_*') for each blues cycle. In bar one a set of four non-terminal, *type-level* chord symbols (see 3.2.2) is probabilistically chosen. It follows that the rest of the top-level 'dec_*' decisions return two non-terminal chord symbols each. In other words, at certain points in the structure the grammar returns a number of productions scheduled as future rewrites or musical events. This approach is similar to the way that Keller and Morrison [27] used probabilistic grammars for the generation of style-abiding jazz melodies. In [27] the grammar controls both rhythmic and melodic structure of melodies and the latter are extended by controlling the length of terminal strings produced (at each point). The points at which top-level choices should be made are determined by the form.

It can be argued that the blues form can be divided into three four-bar sections (for clarity, note that here we take "blues form" to indicate the basic blues form along with its harmonic language (or variations known as rock 'n' roll blues, rock blues or others) and not what is known as jazz blues, *Blues for Alice* form etc.).

For the sake of clarity of harmonic form and form flexibility, in our example further division has been applied (bars 5 to 12). Namely, dec_1, dec_7 and dec_11 can be seen as extended tonics (I), dec_5 as extended subdominant (IV) and dec_9 extended dominant.

Once 'dec_*' non-terminals are rewritten as non-terminal chord symbols, a type-level rule is applied to produce a terminal chord symbol. Rewrites occur in a time-controlled manner. For instance, for the rule shown in Figure 5 the non-terminal 'dec_1' may rewrite as "I IV I I" with probability 0.25 (Figure 5). Next, if the context matches the rule in Figure 4 the tonic (I) in bar 1 will be rewritten as a terminal 'i7' with probability 0.4. Once a terminal has been reached, rewriting stops until the next musical time arrives (the next bar in this case).

3.2.2 Grammar Design

As Section 3.2.1 partly conveys, the grammar can be abstracted as G = (M, T, R, S, P) where we have:

- Start point S → dec_* dec_* dec_* … (form-dependent time-stamped decision points at the beginning of every cycle)
- Non-terminals (non-t) M divided in:
  - Top-level non-t = { dec_*, cad, fin, rec } (see also Sections 3.2.3 to 3.2.5)
  - Type-level non-t = { I, II, III, IV, V }
- Terminals T = { i, i6, i7, iim7, iiim, iiim7, iv, iv6, iv7, v7 }
- Set of rules R
- Set of rewrite probabilities P

Type-level non-t chord symbols function as an intermediate step after top-level non-terminals in order to control the final configuration of each chord according to its position and context (e.g. IV rewrites in either of 'iv', 'iv6' or 'iv7').

For our blues example, the probabilities of production rules are hand-coded based on musical experience, however, we intend that in the future, probabilities will be learnt from style-appropriate corpora. Beyond the above elements, the system accepts a number of form-defining data, i.e. form length, harmonic rhythm (chords per bar), time-placements of top-level decisions in the form ('dec', 'cad'), time signature and an optimal harmonic form (see Section 3.2.5).

Rules R are divided in *timed* (denoted with '_*' in the grammar, Figures 3, 4 & 5) and *general* as well as context-aware or not. Two main divisions of rules R apply to non-terminals:

- *Timed rules* (TR - denoted as '_*' in the grammar) vs. *general rules* (GR).
- and *context-aware* rules vs. *context-free* rules.

As Roads and Wieneke state [26], a grammar described only by rewrite rules is weak for music description and generation of musical structure, unless somehow enhanced. In our case this enhancement relies on the time-specification of certain rules as well as time-based resolution of hierarchical rewrites.

The following sections (3.2.3 to 3.2.6) describe how the music engine responds to the signals received from the prediction engine, i.e. ENDING, RECOVERY and GOAL_REACHED (see Section 3.1).

3.2.3 Cadencing

When 'ENDING' (Section 3.1) becomes true, the music is scheduled to cadence within a number of bars, according to the chosen generated form. In the blues form that has been chosen for the case studies in this paper (see Section 4) two-bar cadences can be placed at every four bars, i.e. bars 3, 7, 11 with the prospect for each cadence to end at bars 5, 9 and 13 (i.e. 1 of the form) respectively (Table 2). This choice was made based on consistency with the 12-bar blues form. In other words, finishing at every two bars or on the even bars of the form would have a syncopated feel, compared to finishing on either of bars 5, 9 or 13.

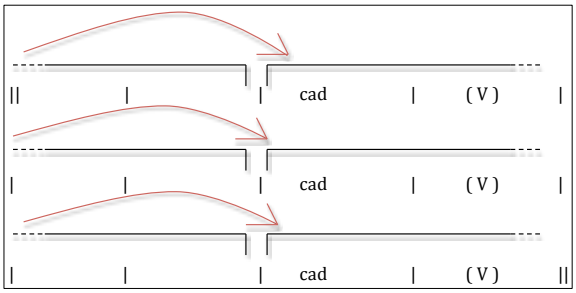


Figure 6. Correspondence of 'ENDING' occurrence with cadence placement.

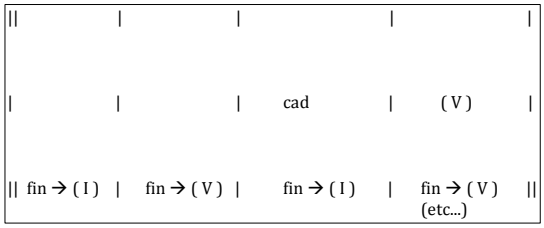


Figure 7. Post-cadencing: 'fin' placed in remaining form.

ENDING occurrence bars	Cadence placement bar	Expected end bar
11 - 2	3	5
3 - 6	7	9
7 - 10	11	13 (1)

Table 2. Potential cadence ('cad') and ending bars.

	Traj. 1	*Traj. 2*	*Traj. 3*
ENDING	21	24, 32	13, 21, 29, 35
RECOVERY	-	31	18, 25, 33
GOAL_REACHED	31	38	41

Table 3: Bars when prediction engine events occur in each trajectory. Bars are numbered linearly irrespective of form.

Depending on when the ENDING signal is received, the cadence is scheduled for the equivalent point in the form as shown in Figure 6 and Table 2. E.g. if the ENDING signal is received within bars three to six, the cadence will be scheduled for bar seven (second row of Table 2). The top-level non-terminal 'cad' probabilistically produces two type-level chords (pre-dominant, dominant) based on the context that precedes the cadence placement bar. It follows that the cadence at bar 11 is placed there despite it being the end of the cycle in order to highlight the ending of the form.

3.2.4 Post-Cadencing

Under the same circumstances, along with scheduling 'cad', the top-level non-terminal 'fin' is placed in every bar after the cadence is finished (Table 2 - column 3, Fig. 7) until the music ends. A micro-grammar of non-terminals 'fin' guarantees that a sequence of V – I chords will be played after the music has cadenced. The purpose of this feature is to prolong the end of music until the goal is reached, by providing some basic harmonic movement that is flexible enough to end within one or two bars after the goal. I.e. once the goal is reached and a cadence has completed, the music stops at the first tonic chord generated.

3.2.5 Recovering

A significant feature of the music engine is its ability to recover from a 'false alarm' from the prediction engine. By that it is meant that the music can be led back into the previously ending form. Unless the goal has been reached, the prediction engine may, theoretically, change from ENDING to RECOVERY (Section 3.1) countless times. Music generation can support this in the following way: Regardless of whether the previously ENDING music has reached its scheduled cadence or not, or even if it is already in a post-cadencing phase, the music engine takes a two-step recovery approach. When a new bar is reached and ENDING has just turned to RECOVERY, an optimum harmonic sequence of the form is applied from one harmonic-rhythm step (i.e. 1 bar) ahead onwards. This aims to re-establish the feel of the form by placing the most "classic" chords at each bar. The optimum harmonic sequence is placed one bar later so that the one gap-bar that connects it with the previous harmonic phase (i.e. pre-cadencing, cadencing or post-cadencing) works as a reconciling chord between the falsely ending harmony and the optimum harmony. In order to preserve non-abrupt harmonic progress, an additional number of timed and general (i.e. non-timed), context-aware rules are defined by the grammar. These rules can support all cases of recovery regardless of the preceding harmonic phase.

The top-level non-terminal for the reconciling gap-bar is 'rec'. As with rule production probabilities, the optimum chord progression has been hand-coded here, but will be statistically modeled in the future.

3.2.6 Stopping

When the goal point is reached and the post-cadencing phase has been entered (even by 1 bar), the grammar productions and music in general stop once a tonic chord is reached, i.e. until a top-level non-terminal 'fin' is rewritten as one of terminals 'i', 'i6' or 'i7'. As mentioned earlier (Section 3.1) reaching the goal point is irreversible, so the 'episode' finishes and recovery is no longer an option.

4. CASE STUDIES

In order to test the effectiveness of the system in generating musical cadences, form recoveries and musical endings on demand, we provided the music engine with data generated by the prediction engine. Computer mouse movement, which served as a proxy for the movement of a virtual participant for our case studies, produced the test trajectories offline. For these case studies of grammar productions, the sequencer was not used but instead a test harness replayed the offline-recorded trajectory data. All three trajectories were performed in a distinctly different way in order to challenge the music engine under different series of events. Trajectory 1 is the smoothest of the three, where the virtual participant approaches the goal point rather directly (Fig. 8) with relatively consistent speed, trajectory 2 significantly diverges from what would be an ideal trajectory towards the goal point with moderate speed variation and, finally, trajectory 3 approaches the goal point but spins in circles before it reaches it, this time with significant speed variation.

Regarding the data received from the prediction engine, the 'ENDING' signal is sent based on a logical conjunction of data levels (i) and (ii) (Section 3.1). Specifically 'ENDING' is sent if (i) the EWMA of minimum remaining time towards an episode end is less than 4 bars' AND if (ii) the participant is approaching the goal point. As explained in Section 3.2.3, 4 bars duration for (i) has been chosen because the blues form ends more naturally in this configuration. Finally, the 'RECOVERY' signal (3.2.5) is sent when 'ENDING' ceases to apply and 'GOAL_REACHED' (3.2.6) occurs as the episode ends.

The recorded prediction engine data were provided to the parser in order to produce the three chord sequences in response to trajectories' events. In order to render audio examples, these chord sequences were translated into MIDI and rendered using Ableton Live[1]. Drums and bass parts were added as pre-programmed MIDI loops, triggered alongside the harmonic parts and – in the case of the bass part – selected based on the grammar-generated harmony. The tempo of music was set at 80 bpm and harmonic rhythm at 1, i.e. one chord per bar.

Decisions of the music engine are made at the beginning of each bar, so Tables 3, 4, 5 and 6 represent the sequence of prediction engine data as the music engine examines them, i.e. at every bar.

5. RESULTS AND DISCUSSION

All music generated from the above input data was successful in cadencing, form recovering and ending the music on demand, in a form-complying manner.

Tables 4, 5 and 6 demonstrate the harmonic states of the blues cycle. Specifically, the column *Grammar states of cycle* in each figure presents bars from 1 through to 12 at the time of corresponding events. '||' denotes the end of music generation. Bars are shown in terms of linear ('*L*') and cycle ('*C*') numbering and each row demonstrates the post-rewrite state of its corresponding bar, i.e. a terminal has been produced for that bar. Arrows from each table row show the trajectory points that trigger each corresponding event, which, in turn, impacts music generation (Tables 4, 5, 6 and Figures 8, 9, 10).

Music for *trajectory 1* is generated according to form rules until bar 21, i.e. bar nine of the form, second time round, when a cadence is introduced. The cadence is placed in bar 11 of the form (second time round) since 'ENDING==true' occurred in bar 9 of the form, i.e. within bars seven to ten (Figure 6). In bar 25, i.e. the first bar of the form in the third cycle, the post-cadence phase is reached and the top-level non-terminal 'fin' populates the form. 'GOAL_REACHED' occurs in bar 31 (or seven in the third cycle) where a tonic has been produced, so grammar productions and music stop there.

In *trajectory 2* we have a similar development until bar 30. As shown in Table 5 at bar 29 a post-cadencing phase has been reached ('fin') while previously the cadence at bar three of the form has been scheduled since bar 24. When bar 31 is reached, a recovery bar is scheduled for the next bar, i.e. bar eight of the form. Along with it, optimum chords are placed in the remaining bars. Subsequently, at bar 32 a cadence is scheduled again for bar 11 of the form and post-cadencing begins at bar 37. The goal is reached at bar 38 and music generation continues only for one more bar (39) when a tonic is produced.

In a similar way *trajectory 3* schedules a cadence at bar 24 (for the third bar of the form), post-cadences at bar 17 and schedules recovery at bar 18 for the next bar (seventh of the form). At bar 25 it is interesting to note that post-cadencing is cancelled by a recovery at that point. So instead of a 'fin' population (for bars two to twelve of the form), the optimum chords populate bars three to twelve with a 'reconciling' chord ('rec') at bar two of the form. The same happens at bar 33 when post-cadencing would be expected at bar ten of the form. However, post-cadencing coincides with recovery again and recovery overrules as expected. In addition, it is worth mentioning that at bar 35, when a cadence is scheduled for bar three of the form, 'rec' is still present at bar ten of the cycle. This is a leftover symbol from the preceding recovery (bar 33) and is cancelled by 'fin' once the post-cadence phase is reached (bar 41). Finally, post-cadencing (bar 41) coincides with GOAL_REACHED and since the 'current' chord is a tonic (form bar five) the music stops.

Generally, as Tables 4, 5 and 6 show, the positioning of top-level non-terminals (*dec*, *cad*, *fin, rec*) is, in most cases, scheduled in advance. Thus, the non-terminal is not rewritten until its musical time (i.e. bar) has arrived.

6. CONCLUSIONS AND FUTURE WORK

In summary, FATE is a real-time system design that generates music in a form-aware manner and can respond to environment data to cadence, recover and end musical form on demand. It aims to generate music in a way that retains the flexibility of the short-form approaches like

[1] At this stage translation into MIDI was made by hand although it will be automatic in the future.

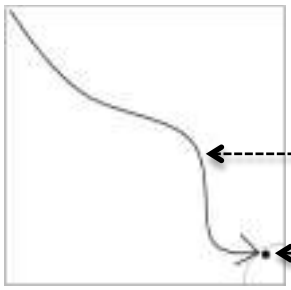


Figure 8. Trajectory 1.

Trajectory 1			
Events	*Bars*		*Grammar states of cycle*
	L	*C*	
No significant input	20	8	i i6 v7 i7 iv6 iv6 i7 i6 dec v7 dec v7
ENDING	21	9	i i6 v7 i7 iv6 iv6 i7 i6 v7 IV **cad** v7
Post-cadencing	25	1	i6 fin fin fin fin fin fin fin fin fin fin fin
GOAL_REACHED	31	7	i6 v7 i6 v7 i6 v7 i7 fin fin fin fin fin
Stopping	31	7	i6 v7 i6 v7 i6 v7 i7 **||** fin fin fin fin fin

Table 4. Events, bars of occurrence and current cycle state at each bar for trajectory 1.

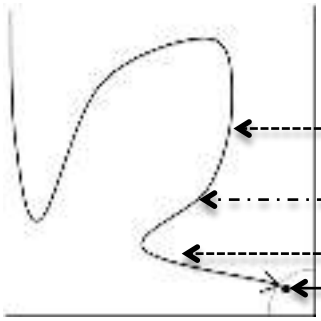


Figure 9. Trajectory 2.

Trajectory 2			
Events	*Bars*		*Grammar states of cycle*
	L	*C*	
No significant input	23	11	i7 v7 i7 i7 iv7 iv7 iiim7 i6 v7 iv7 i7 V
ENDING	24	12	i7 v7 **cad** i7 iv7 iv7 iiim7 i6 v7 iv7 i7 V
Post-cadencing	29	5	i7 v7 iv7 v7 i7 fin fin fin fin fin fin fin
RECOVERY	31	7	i7 v7 iv7 v7 i7 v7 i7 **rec** V IV I I
ENDING	32	8	i7 v7 iv7 v7 i7 v7 i7 i7 V IV **cad** I
Post-cadencing	37	1	i7 fin fin fin fin fin fin fin fin fin fin fin
GOAL_REACHED	38	2	i7 v7 fin fin fin fin fin fin fin fin fin fin
Stopping	39	3	i7 v7 i7 **||** fin fin fin fin fin fin fin fin fin

Table 5. Events, bars of occurrence and current cycle state at each bar for trajectory 2.

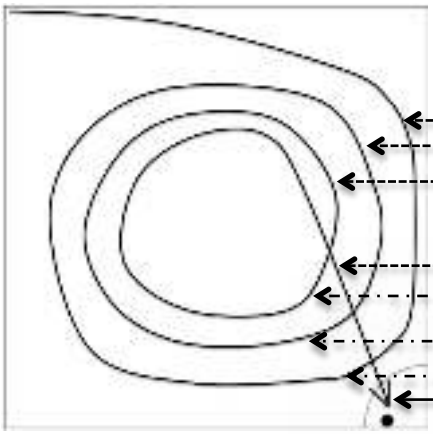


Figure 10. Trajectory 3

Trajectory 3			
Events	*Bars*		*Grammar states of cycle*
	L	*C*	
No significant input	12	12	i7 iv7 i7 v7 iv7 iv6 i7 i6 v7 iv7 i7 v7
ENDING	13	1	i7 IV **cad** V dec iv6 dec i6 dec iv7 dec v7
Post-cadencing	17	5	i7 iv7 iim7 v7 i7 fin fin fin fin fin fin fin
RECOVERY	18	6	i7 iv7 iim7 v7 i7 v7 **rec** I V IV I I
ENDING	21	9	i7 iv7 iim7 v7 i7 v7 i7 i6 v7 IV **cad** I
Post-cadencing	25	1	(cancelled by RECOVERY–see next row)
RECOVERY	25	1	i6 **rec** I I IV IV I I V IV I I
ENDING	29	5	i6 iv7 i7 i7 iv7 IV **cad** I V IV I I
Post-cadencing	33	9	(cancelled by RECOVERY–see next row)
RECOVERY	33	9	i6 iv7 i7 i7 iv7 iv6 iim7 v7 i7 **rec** I I
ENDING	35	11	i6 iv7 **cad** i7 iv7 iv6 iim7 v7 i7 (rec) I I
Post-cadencing	41	5	i6 iv7 iim7 v7 i7 fin fin fin fin fin fin fin
GOAL_REACHED	41	5	i6 iv7 iim7 v7 i7 fin fin fin fin fin fin fin
Stopping	41	5	i6 iv7 iim7 v7 i7 **||** fin fin fin fin fin fin

Table 6. Events, bars of occurrence and current cycle state at each bar for trajectory 3.

procedural note/chord-level generation and dynamic layers, while managing larger-scale form.

The design is divided in two: a prediction engine that captures EPE episode data and estimates the remaining time of that episode and a music engine that generates music in real-time. The prediction engine uses an EWMA of a participant's speed (among other calculations) to estimate the remaining time. The music engine uses a stochastic, context-sensitive grammar that allows for hierarchical grammar rewrites at specific times of the musical form. Currently we address the non-abrupt musical termination of episodes.

The parser enables the programmer/computer composer (length, top-level time-specific decision points, rewrite rules) in order to generate the harmony of a medium-(length, top-level time-specific decision points, rewrite rules) in order to generate the harmony of a medium-length chorus-based form such as 12-bar blues and jazz standards.

Future work will address a number of open challenges:
1. Grammatical production probabilities as well as the optimum harmonic sequence will be learnt from a corpus.
2. Longer 'reconciling' musical periods will be applied to test their impact on musical smoothness of harmonic recovery and how effectively the form is re-established. Also, the location of the reconciling period will be varied.
3. The algorithm will be further developed to support generation and non-abrupt ending of musical forms that are longer and less repetitive than the medium-length chorus-based forms that are currently supported.
4. The algorithm will be further developed in order to generate musical transitions between two musical sections rather than simply ending and recovering one.

5. The system will be applied for musical accompaniment of a real-world interactive theatrical installation.

Acknowledgments

This work is funded by the UK Engineering and Physical Sciences Research Council through the VEIV Doctoral Training Centre at UCL [grant number: EP/G037159/1] and the Alexander S. Onassis Foundation. We are grateful for the partnership of Penny Dreadful Productions in this work. Data in support of this paper is available at DOI: 10.14324/000.ds.1503636

7. REFERENCES

[1] B. Magerko, W. Manzoul, M. Riedl, A. Baumer, D. Fuller, K. Luther, C. Pearce, An Empirical Study of Cognition and Theatrical Improvisation – creativity '09, 2009, Berkeley, California, USA.

[2] Cirque du Soleil Theme Park, https://www.cirquedusoleil.com/en/press/news/2014/grupo-vidanta-nueva-vallarta-entertainment-park-12[2], 2014 (announcement).

[3] DisneyQuest®, https://disneyworld.disney.go.com/entertainment/disney-springs/disney-quest-indoor-interactive-theme-park/, 1998 – present.

[4] Blast Theory, Desert Rain, www.blasttheory.co.uk/, 2000.

[5] M. Niemann, Five Minutes, www.fiveminutes.gs, 2014.

[6] K. Collins, Game sound: An Introduction to the History, Theory, and Practice of Video Game Music and Sound Design, MIT Press, 2008.

[7] S. Williams, Music in the Theatre, The Oxford Companion to Theatre and Performance, ed. by D. Kennedy, Oxford University Press, 2010.

[8] S. Benford, G. Giannachi, B. Koleva, T. Rodden, "From Interaction to Trajectories: Designing Coherent Journeys Through User Experiences", CHI 2009, Boston, USA.

[9] J.D. Fernandez, F. Vico,"AI Methods in Algorithmic Composition: A Comprehensive Survey", Journal of Artificial Intelligence Research 48 513-582, 2013.

[10] M. Muller, J. Driedger, "Data-driven soundtrack generation, Multimodal Music Processing", Dagstuhl Follow-Ups - vol. 3, Multimodal Music Processing, ed. by M. Muller, M. Goto, M. Schedl, ISBN 978-3-939897-37-8, 2012.

[11] Hello Games, No Man's Sky, http://www.no-mans-sky.com/, 2016.

[12] Thatgamecompany, Journey, http://thatgamecompany.com/games/journey/, 2012.

[13] A. R. Brown, R. Wooller, K. Thomas, "The Morph Table: A collaborative interface for musical interaction", Proc. Australasian Computer Music Conference 2007, Lefkada, Greece.

[14] M. O. Jewell, Motivated Music: Automatic Soundtrack Generation for Film, PhD Thesis, University of Southampton, 2007.

[15] E. Vane, W. Cowan, "A computer-aided soundtrack composition system designed for humans", International Computer Music Conference, 2007.

[16] D. Goldmark, Pixar and the Animated Soundtrack, The Oxford Handbook of New Audiovisual Aesthetics, ed. by J. Richardson, C. Gorbman, C. Vernallis, Oxford University Press, 2010.

[17] A. Hazzard, Guidelines for Composing Locative Soundtracks, PhD Thesis, University of Nottingham, 2014.

[18] S. P. Langston, (201) 644-2332 or Eedie & Eddie on the wire: An experiment in music generation, Bell Communications Research Morristown, NJ, 1986.

[19] Lucasfilm Games, Ballblazer, http://www.mobygames.com/game/ballblazer, 1984.

[20] G. Whitmore, Adaptive Audio Now! A Spy's Score: A Case Study for No One Lives Forever, DirectX 9 Audio Exposed: Interactive Audio Development, ed. by T. M. Fay, S. Selfon, T. J. Fay, Plano, Texas: Wordware Publishing, 2004.

[21] Microsoft, Russian Squares, https://www.youtube.com/watch?v=a-ZyozBeUDg, 2002.

[22] Y. Ioannides, [.talktome]: adaptive/dynamic audio prototyping for video games, https://cycling74.com/project/talktome-adaptivedynamic-audio-prototyping-for-video-games/#.V4pq-TaqTIw, 2012.

[23] B. Eno, Spore, http://www.spore.com, Electronic Arts, 2009.

[24] Strange Loop Games, Simcell, http://www.strangeloopgames.com/education/, 2012.

[25] R. Wooller, Techniques for automated and interactive note sequence morphing of mainstream electronic music, Queensland University of Technology, PhD Thesis, 2007.

[26] C. Roads, P. Wieneke, "Grammars as representations for music", Computer Music Journal, vol.3, no.1, 1979.

[27] R. M. Keller, D. R. Morrison, "A grammatical approach to automatic improvisation", Proc. Sound and Music Computing Conference, 2007, Lefkada, Greece.

[2] All URLs shown were last accessed on 15/07/2016.

VIRTUAL RECONSTRUCTION OF AN ANCIENT GREEK PAN FLUTE

Federico Avanzini, Sergio Canazza, Giovanni De Poli, Carlo Fantozzi, Edoardo Micheloni, Niccolò Pretto, Antonio Rodà
Dept. of Information Engineering
University of Padova
name.surname@unipd.it

Silvia Gasparotto
University IUAV of Venice
silvia.gasparotto@gmail.com

Giuseppe Salemi
Dept. of Cultural Heritage
University of Padova
giuseppe.salemi@unipd.it

ABSTRACT

This paper presents ongoing work aimed at realizing an interactive museum installation that aids museum visitors learn about a musical instrument that is part of the exhibit: an exceptionally well preserved ancient pan flute, most probably of greek origins. The paper first discusses the approach to non-invasive analysis on the instrument, which was based on 3D scanning using computerized tomography (CT scan), and provided the starting point to inspect the geometry and some aspects of the construction of the instrument. A tentative reconstruction of the instrument tuning is then presented, which is based on the previous analysis and on elements of theory of ancient Greek music. Finally, the paper presents the design approach and the first results regarding the interactive museum installation that recreates the virtual flute and allows intuitive access to several related research facets.

1. INTRODUCTION

Preservation of documents is usually categorized into *passive* preservation, meant to protect the original documents from external agents without alterations, and *active* preservation, which involves the data transfer from the analogue to the digital domain. The traditional "preserve the original" paradigm has progressively shifted to the "distribution is preservation" idea of digitizing the content and making it available in digital libraries [1].

In recent projects [2–4], some of the present authors have proposed to transpose these categories to the field of physical artifacts and musical instruments. In this context, passive preservation is meant to preserve the original instruments from external agents without altering the components, while active preservation involves a new design of the instruments using new components or a virtual simulation of the instruments. Specifically, the approach adopted here amounts to developing virtual counterparts in the digital domain, which retain as much as possible the characteristics of the original instruments.

Providing new means of interaction with instruments that are otherwise not accessible means also proving new means to access them on a wide scale, particularly in museum exhibits, where presenting artifacts to the general public is a complex task, because of their multi-faced nature. Interactive museum installations can increase the engagement and participation of visitors [5], and enforce new forms of learning where visitors observe and perceive artifacts by means of multiple senses and control them through movements [6], a more natural form of learning than one based on symbols, which are not accessible to perception. Ultimately, applying new technologies to interactive museum installations can create stronger consensus and interest for the preservation of cultural heritage [7].

In addition to permanent installations, museum exhibits can also benefit from mobile applications that exploit unprecedented multimedia and multisensory capabilities offered by smartphones and tablets, which are endowed with a wide range of sensors and input devices, as well as non-negligible computing power (multi-core CPUs, augmented with specialized accelerators for 3D graphics and signal processing operations). Consequently mobile devices are finding significant applications in the virtual reconstruction of environments [8] and physical objects [9], allowing the development of *skeuomorphic* user interfaces, i.e. interfaces that leverage on the appearance and behavior of physical artifacts. In this context, apps for musical cultural heritage are a particularly interesting domain.

This paper presents current results of an ongoing research project, which combines a team of researchers in such field as archaelogy, 3D scanning and modeling, and sound and music computing, around a unique artistic artifact: an exceptionally well preserved ancient pan flute, most probably of greek origins, recovered in Egypt in the 1930's and currently exhibited in the Museum of Archaeological Sciences and Art at the University of Padova. Before being included in the permanent exhibit, the flute underwent a major restoration programme for consolidation and (passive) preservation, as shown in Fig 1. Details about the history of the artifact, the place and circumstances of its recovery, as well as related literary and iconographic references in the Greek-Roman world, are provided in a previous publication [3].

The final goal of the project is to develop an interactive museum installation that virtually re-creates the instrument, and communicates different aspects related to

Figure 1. The restored pan flute (frontal and posterior views).

history, iconography, acoustics, musicology, etc., as well as the research carried out during the project. Achieving this goal requires truly multidisciplinary methodologies as it entails (i) studying the history and iconography of pan flutes, with a focus on Classical Greece; (ii) analyzing the geometry, construction, age and geographical origin of this artifact through non-invasive techniques such as 3D scanning and materials chemistry; (iii) studying its acoustics, timbre, and tuning, also by combining physics with elements of ancient Greek music theory; (iv) designing interactive installations that recreate a virtual flute allowing intuitive access to all these facets. These concepts may be summarized in a single "mission statement": we want to bring back to light archeological remains, but also to bring them back to life, with the aid of technology.

The remainder of the paper is organized as follows. Section 2 presents the results of a campaign of non-invasive measurements performed on the flute by means of CT scanning. These measures are the starting point for the analysis of the tuning of the flute, which is discussed in Section 3. Finally, Section 4 presents the interactive applications that are being developed on the basis of these results: specifically an interactive museum installation is discussed in Section 4.1, while a mobile application is presented in Section 4.2.

2. MEASUREMENTS

It is known that the internal lengths of the pipes are reduced by carefully increasing the thickness of the closed ends through the addition of wax or propolis, in order to fine-tune fundamental frequencies [10]. Despite the restoration of the ancient pan flute, some pipes are still partially obstructed, therefore the interior of these pipes is not completely visible and not directly inspectable. In a previous work [3] a preliminary estimation of pipe lengths was obtained from external measures taken on a laser-scanned 3D model. In order to refine these measurements, computerized tomography (CT) scan was used here. Specifically, in order to determine fundamental frequencies of the pipes, two measures were estimated: internal length and internal diameter.

The three dimensional image of the interior of the instrument is obtained with a GE LightSpeed VCT 64 Slice CT scan. The scanning was then read with the open-source software Horos, a medical image viewer which also provides tools to extract reliable measures from the CT scan.

In order to browse inside the three-dimensional image, and to perform precise measures, the operators alternated two different views: a 3D MultiPlanar Reconstruction (MPR) and a 2D orthogonal MPR. Figures 2a, 2b, and 2c show the latter view and the three orthogonal planes, defined as axial, coronal and sagittal, respectively.

Since some parts are damaged or corrupted, a total of eighteen measurements for every pipe were collected, with the goal of obtaining more robust estimates.

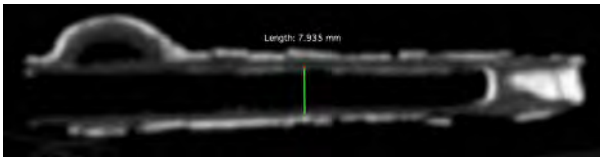
With regard to pipe lengths, six measures were extracted from axial and coronal planes. Pipe openings are not straight, they are slightly u-shaped at one side in order to provide an embouchure to the player (the opening shapes can be observed in Fig. 1, posterior view): therefore, the difference between the maximum and the minimum point of the opening was measured on both planes. Moreover, the internal shapes of the closed pipe ends are also not straight: therefore, for each plane, the maximum and minimum internal lengths were measured (Fig. 2b shows an example of measuring the maximum length of a pipe in the coronal plane).

With regard to pipe diameters, twelve measures were collected. One measure was taken in the axial plane and a second one in the coronal plane, whereas two measures were taken in the sagittal plane (one for each axis of the pipe, see Fig. 2c), because pipe sections are oval-shaped rather than circular. These four measures were repeated at three different levels: near the opening of the pipe, at the mid point and near the closed end (Fig. 2a shows a diameter measure at the mid point in the axial plane).

The measurement process highlighted several issues that required some subjective interpretations by the operators. The longest pipe, for example, is broken and curved, thus a specific tool of 3D Curved-MPR was used that, given a set of reference points on the curve, virtually straighten the pipe, providing a more usable view for the correct measurement. In other cases, the presence of obstructing materials impaired a correct evaluation of the internal surface of the pipe. This is the main reason why the authors chose to take redundant measurements. Furthermore, for the shortest pipes in was not possible to measure directly the position of the openings because these pipes are more heavily damaged. Consequently, opening positions were estimated from the neighboring pipes. The error that mostly impact the measures was the CT scan resolution: every voxel (volumetric pixel) is isometric and it measures $0,625$ mm. All these difficulties affected the accuracy of the measures, however using redundant measurements provided a range for a plausible estimation of lengths and diameters.

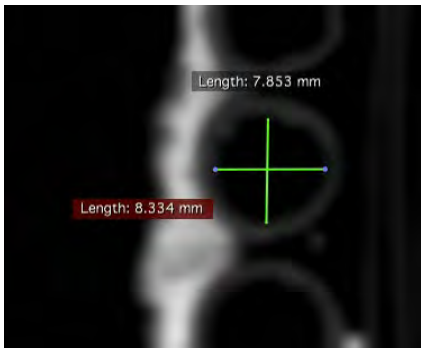
3. TUNING

The measurements of the internal length and diameter of the pipes were used to estimate their fundamental frequencies, under the assumption of ideal open-closed cylindrical



(a)

(b)



(c)

Figure 2. Views from the CT scan; (a) example of diameter measurement on the axial plane; (b) example of length measurement on the coronal plane; (c) example of diameter measurement on the sagittal plane.

pipes:

$$f = \frac{c}{4(l_{\text{int}} + \Delta l)} \quad \text{Hz}, \tag{1}$$

where c is the sound velocity, l_{int} is the internal pipe length, and $\Delta l \sim 0.305 d_{\text{int}}$ is the length correction at the open end, which is proportional to the internal pipe diameter d_{int} [11]. As the measurements are affected by the errors reported in the previous section, for each pipe we considered the minimum and maximum values of internal length and diameter.

Then, in order to take into account the effects of error propagation, an interval of values was calculated for each pipe as an estimate of the fundamental frequency: in particular, f_{min} was calculated from Eq. (1) using the maximum values of length and diameter, whereas f_{max} was calculated from corresponding the minimum values (see Table 1).

pipe	$f_{min}[Hz]$	$f_{max}[Hz]$
1	638.7	649.7
2	677.2	700.7
3	753.6	773.5
4	843.1	874.4
5	928.3	974.7
6	1010.1	1041.3
7	1142.2	1184.3
8	1283.2	1346.4
9	1389.6	1438.2
10	1538.3	1602.0
11	1721.8	1758.1
12	1901.4	1957.3
13	2128.4	2205.1
14	2292.9	2499.7

Table 1. Fundamental frequencies (min and max) estimated for each pipe starting from the measurements taken from the CT scan.

It is interesting to verify whether these frequency ranges are compatible with predictions derived from music theory. According to theorists [12], the ancient Greek music system was based on the *tetrachord*, i.e a group of four notes (often associated to the four strings of the lyre or the kithara) where the ratio between the pitches of the fourth note and the first note is equal to $4:3$, namely a perfect fourth.

Figure 3 shows the pitch ratios calculated as $f(n+3)/f(n)$ for $n = 1, 2, ..., 11$, where $f(n)$ is the fundamental frequency of the n^{th} pipe. Due to error propagation, for each pair of pipes a range of values (reported with the vertical lines) was obtained. Comparing the ranges with the horizontal line representing the $4:3$ ratio (dot-dashed line), it is possible to see that all the intervals are compatible with the tetrachord definition.

It is known that the tetrachord is subdivided into three pitch intervals that can have various configurations. In particular, three *genera* can be distinguished: diatonic, chromatic, and enharmonic. As an example, the diatonic tetrachord is characterized by intervals that are less than or equal to half the total interval of the tetrachord. Usually, this tetrachord begins with one small interval followed by two larger intervals, corresponding approximately to a tone $(9:8)$.

Figure 4 shows the pitch ratios between adjacent pipes, i.e $f(n+1)/f(n)$: it is possible to recognize some intervals that are compatible with a tone $(9:8)$ and other smaller intervals compatible with what some theorists call *diesis*, corresponding to the ratio $256:234$. Two tetrachords can be joined following two different schemes, called *synaphē* (conjunction), when the top note of the lower tetrachord corresponds to the bottom note of the higher one, and *diazeuxis* disjunction, when there is an interval of a tone between the tetrachords. Observing the sequence of intervals of Figure 4, some joint tetrachords can be recognized: e.g., the pitch of the first eight pipes are compatible with two disjoint tetrachords, as represented in Figure 5.

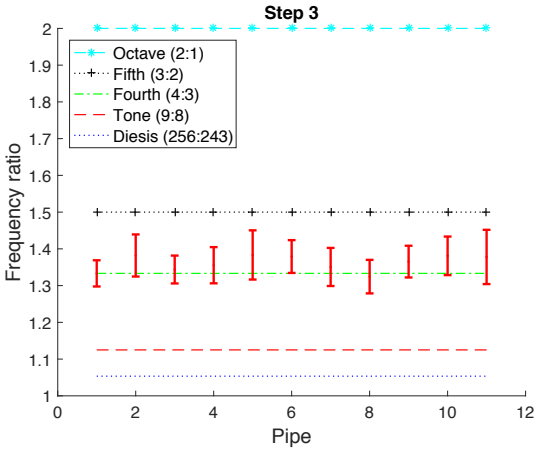


Figure 3. Pitch ratios calculated as $f(n+3)/f(n)$, where $f(n)$ is the fundamental frequency of the n^{th} pipe. The horizontal lines correspond to the basic theoretic intervals.

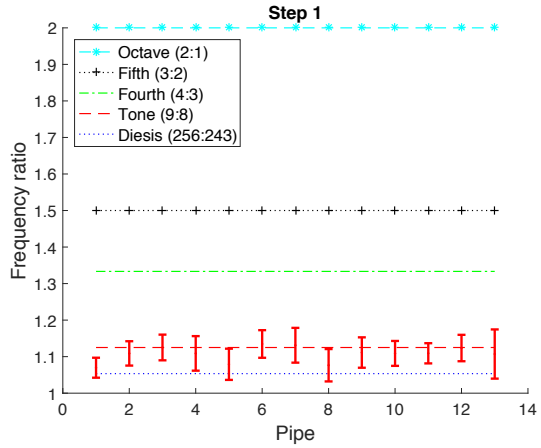


Figure 4. Pitch ratios calculated as $f(n+1)/f(n)$, where $f(n)$ is the fundamental frequency of the n^{th} pipe. The horizontal lines correspond to the basic theoretic intervals.

4. VIRTUAL RECONSTRUCTIONS

4.1 Museum installation

An interactive museum installation is being developed, and will be included in the permanent exhibit of the Museum of Archaeological Sciences and Art at the University of Padova. The installation will allow visitors to have a direct multimodal experience of the flute sound, and to access easily and intuitively to all the related information [5–7].

The installation is physically composed of two parts connected together into a single structure (see Fig. 6). These parts are designed to facilitate two different interactions: the first one (left side) is concerned with the sound, while the second one (right side) provides visual information about the musical instrument. In particular, the first part is designed to symbolically represent the 14 pipes of the flute through 14 cuts of different lengths made on the top of the furniture. The visitor can "play" the installation by blowing into the holes at the bottom of the cuts, and listening to the resulting sound from the corresponding pipe, additionally visualizing the interaction through the lighting of LED stripes placed inside the cuts.

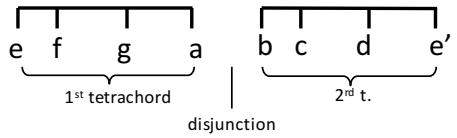


Figure 5. Schema of two disjoint tetrachords, compatible with the pitch of pipes 1-8. The letters used to represent the notes does not correspond to modern pitches.

Figure 6. The first realization of the multimedia installation.

In order to detect the air flow generated by the user, every hole is equipped with a printed circuit board (PCB) that includes a small condenser microphone and an amplification circuit, forming an array of 14 microphones [13, 14]. All the sensors are connected to an Arduino board, that estimates the amplitude envelope and the energy of the flow signal with appropriate processing, and traslates these features into MIDI messages. These in turn are sent as input to an audio interface that controls a sampler of the pan flute through Ableton Live.

The second part consists of a touch display that allows visitors to read some informations expressed through texts, photos, videos, 3D images, and illustrations that tell about different aspects of the pan flute.

The design of the installation required to take into account several elements, including the peculiarities of the architectural context of the "Liviano" Building (designed by Giò Ponti in 1932), where the exhibit is hosted. For this reason the shape and the color of the installation were chosen to be in style with the modernist architecture of the museum. Other elements driving the design were the robustness and durability of the installation, as well as the user experience.

The interaction was designed to make information as intuitively accessible as possible. Specifically, the navigation bar is composed of five different sections: *Myth, Flute, History, 3D, Sound*, all of which are represented by a hand drawing sketch and a brief description.

All the sections are characterized by a navigation bar at the bottom of the page that enables the fruition of the contents. The *Myth* and *Flute* sections contain simple textual information and images. In the *History* section, the navi-

gation bar represents a timeline divided by key points for exploring the main iconographic sources, that will be contextualized by a map, a text and an image. The *3D* section provides two tools to interactively explore (i) a texturized 3D model of the flute (obtained from a high resolution laser scan) and (ii) the CT scan discussed earlier. The latter tool uses the navigation bar to explore points of interest in the three dimensional image.

The last section is relative to the *Sound* of the pan flute. Here the user can listen to a note relative to the pipe by touching one of 14 stylized segments. This section is strictly connected with the first part of the installation: there is only one speaker available, thus the two parts work with a mutual exclusion mechanism.

4.2 Mobile application

In addition to the installation, an Android app is being developed to access a subset of the information available in the museum, and, chiefly, a second virtual reconstruction of the pan flute. The virtual musical instrument can be played by moving the mobile device below the mouth, like an actual pan flute, and blowing at the device itself.

The reconstruction is less accurate than one that uses custom hardware. However, an app for commodity mobile devices has the advantage of targeting a much wider public, as it can be freely installed by anyone on her/his smartphone. Moreover, the app virtually takes the flute out of the museum: through the app, anyone can interact with the flute while at home and anywhere. As a consequence, the app is a definitely effective communication vessel for the activities in the project, and it fosters cultural dissemination via informal learning. Furthermore, to the best of our knowledge our app is the first virtual reconstruction of a hole-less wind instrument on commodity mobile devices that aims at a natural interaction with the instrument itself. This is in contrast with currently available simulators: the easy choice of selecting the note to play by touching the screen is reasonable for ocarinas [15], but it is unnatural if adopted – as it is nowadays – for instruments without holes such as harmonicas [16] or pan flutes.

While the virtual flute is being played, the mobile device displays a full-screen picture of the instrument and is held below the mouth of the user: the app infers the note to be played by tracking the movements of the mobile device, hence determining which virtual pipe is displayed right below the mouth. The note is played when the user blows. Blow detection is quite plain: the position of the microphones vary from device to device and hinders a reliable detection of blow intensity, hence we opted for a simple threshold detector. On the contrary, motion tracking is nontrivial and combines information from several device sensors: (a) the orientation of the device in space is detected via the accelerometer and gyroscope; (b) fast (hence, wide) movements of the device are measured via the accelerometer; (c) slow (hence, small) movements of the device are estimated by tracking the user's position via the front camera of the device.

Data from the accelerometer are processed with a Kalman filter, which improves the position estimate; misalignments or rotations of the device with respect to the direction of motion are also compensated. However, when the movements are small and acceleration is low, noise in accelerometer data and error in the orientation estimate cause drift, and a different approach is necessary.

As soon as the measured acceleration falls below a predefined threshold, the app switches to a camera-based motion estimation algorithm. While the flute is being played, the front camera frames part of the user's chin, cheek, neck, and torso (an example is shown in Figure 7). The resulting image is feature-rich enough to track the relative movement of the user with respect to the device. The reasonable assumption is made that the image does not contain independently moving objects, hence image alignment techniques can be adopted [17]. It must be noted that switching to the camera-based estimation strategy has the side benefit of resetting the position error accumulated while integrating accelerometer data. It must also be remarked that the camera cannot replace other kinds of sensors, such as the accelerometer and gyroscope, during fast motion, because in this scenario images are too blurry to be useful.

5. CONCLUSIONS

The main foreseen developments in the short term concern the tuning analysis and the virtual realizations. Regarding the first point, more reliable estimates may be obtained by exploiting more constraints about possible tunings based on elements of music theory of ancient Greek music. Regarding the second point, a thorough validation of the usability of the virtual realizations will be conducted with experimental subjects.

In the mid term the work presented here is expected to produce several developments. One is sound synthesis of the pan flute by means of physical models instead of sampling (as in the current realization), in order to increase the interactivity of the instrument. This is also an interesting research topic *per se*, since to our knowledge only one previous study on sound synthesis of the pan flute is available in the literature [18]. A very high resolution 3D model can also be exploited for computationally intensive acoustic simulations (e.g., based on finite differences or finite elements), which in turn could aid sound synthesis [19].

As far as virtual instrument reconstructions are concerned, an intesting development would be using a 3D print of the flute model (possibily "digitally restored" before printing), which can be sensorized and used as a tangible interface that recreates the physicality of the original instrument. This scenario is in line with current research on 3D printing of musical instruments [20]. However it raises various pratical concerns regarding possible uses in a museum exhibit, particularly about durability and hygienic issues.

Being the pan flute a primeval instrument which is widespread in different cultures worldwide, the impact of this research goes beyond this particular exemplary. We believe that the proposed "active preservation" approach can be applied to other ancient musical instruments.

Acknowledgments

This work was supported by the research project *Archaeology & Virtual Acoustics*, University of Padova, under grant no. CPDA133925.

(a) (b) (c)

Figure 7. Mobile application: tracking small movements of the mobile device via the built-in front camera. (a): reference frame, with a yellow rectangle indicating the central portion of the frame whose motion is actually tracked. (b): a later frame, where the user roughly appears to have moved to the right (actually, it is the mobile device that is moving while the user plays the virtual flute, and the motion is not a pure translation). (c): the algorithm computes the motion, thus allowing to project the central portion of (a) onto (b) and, ultimately, to determine which virtual pipe is below the mouth of the user.

6. REFERENCES

[1] E. Cohen, "Preservation of audio in folk heritage collections in crisis," *Proc. Council on Library and Information Resources*, pp. 65–82, 2001.

[2] F. Avanzini and S. Canazza, "Virtual analogue instruments: an approach to active preservation of the studio di fonologia musicale," in *The Studio di Fonologia - A Musical Journey*, M. Novati and J. Dack, Eds. Milano: Ricordi (MGB Hal Leonard), June 2012, pp. 89–108.

[3] F. Avanzini, S. Canazza, G. De Poli, C. Fantozzi, N. Pretto, A. Rodà, I. Angelini, C. Bettineschi, G. Deotto, E. Faresin, A. Menegazzi, G. Molin, G. Salemi, and P. Zanovello, "Archaeology and virtual acoustics. a pan flute from ancient Egypt," in *Proc. Int. Conf. Sound and Music Computing (SMC2015)*, Maynooth, July 2015, pp. 31–36.

[4] S. Canazza, C. Fantozzi, and N. Pretto, "Accessing tape music documents on mobile devices," *ACM Trans. on Multimedia Comput., Commun. Appl.*, vol. 12, no. 1s, p. 20, 2015.

[5] N. Simon, *The participatory museum*. Museum 2.0, 2010, Creative Commons.

[6] N. Levent and A. Pascual-Leone, *The Multisensory Museum: Cross-Disciplinary Perspectives on Touch, Sound, Smell, Memory, and Space*. Rowman & Littlefield, 2014.

[7] S. Styliani, L. Fotis, K. Kostas, and P. Petros, "Virtual museums, a survey and some issues for consideration," *J. of Cultural Heritage*, vol. 10, no. 4, pp. 520–528, 2009.

[8] J. Thomas, R. Bashyal, S. Goldstein, and E. Suma, "Muvr: A multi-user virtual reality platform," in *Proc. IEEE Virtual Reality (VR2014)*, 2014, pp. 115–116.

[9] M. J. Figueiredo, P. J. Cardoso, C. D. Goncalves, and J. M. Rodrigues, "Augmented reality and holograms for the visualization of mechanical engineering parts," in *Proc. Int. Conf Information Visualisation (IV2014)*, 2014, pp. 368–373.

[10] E. Civallero, *Introducción a las flautas de pan*, 1st ed., Madrid, 2013, Creative Commons.

[11] N. H. Fletcher and T. D. Rossing, *The physics of musical instruments*. New York: Springer-Verlag, 1991.

[12] S. Hagel, *Ancient Greek music: a new technical history*. Cambridge University Press, 2009.

[13] D. Salvati, S. Canazza, and A. Rodà, "A sound localization based interface for real-time control of audio processing," in *Proc. of the 14th Int. Conf. on Digital Audio Effects*, 2011, pp. 177–184.

[14] D. Salvati, A. Rodà, S. Canazza, and G. L. Foresti, "A real-time system for multiple acoustic sources localization based on isp comparison," in *Proc. Conf. on Digital Audio Effects*, 2010, pp. 201–208.

[15] G. Wang, "Ocarina: Designing the iPhone's magic flute," *Computer Music Journal*, vol. 38, no. 2, pp. 8–21, 2014.

[16] M. Vahida. (2016) Harmonica musical instrument. [Online]. Available: https://play.google.com/store/apps/details?id=masih.vahida.and_saz_harmonica_in_app_purchase_en

[17] S. Baker and I. Matthews, "Lucas-Kanade 20 years on: A unifying framework," *Int. J. of Computer Vision*, vol. 56, no. 3, pp. 221–255, 2004.

[18] A. Czyżewski, J. Jaroszuk, and B. Kostek, "Digital waveguide models of the panpipes," *Archives of Acoustics*, vol. 27, no. 4, pp. 303–317, 2002.

[19] S. Bilbao, B. Hamilton, A. Torin, C. Webb, P. Graham, A. Gray, K. Kavoussanakis, and J. Perry, "Large scale physical modeling sound synthesis," in *Proc. Stockholm Musical Acoustics Conf. (SMAC 2013)*, Stockholm, Aug. 2013, pp. 593–600.

[20] A. Zoran, "The 3d printed flute: Digital fabrication and design of musical instruments," *J. New Music Res.*, vol. 40, no. 4, pp. 379–387, 2011.

SKETCHING SONIC INTERACTIONS BY IMITATION-DRIVEN SOUND SYNTHESIS

Stefano Baldan, Stefano Delle Monache, Davide Rocchesso, Hélène Lachambre
Iuav University of Venice, Iuav University of Venice, Iuav University of Venice, Genesis Aix-en-Provence
stefanobaldan@iuav.it, sdellemonache@iuav.it,
roc@iuav.it, helene.lachambre@genesis.fr

ABSTRACT

Sketching is at the core of every design activity. In visual design, pencil and paper are the preferred tools to produce sketches for their simplicity and immediacy. Analogue tools for sonic sketching do not exist yet, although voice and gesture are embodied abilities commonly exploited to communicate sound concepts. The EU project SkAT-VG aims to support vocal sketching with computer-aided technologies that can be easily accessed, understood and controlled through vocal and gestural imitations. This *imitation-driven* sound synthesis approach is meant to overcome the ephemerality and timbral limitations of human voice and gesture, allowing to produce more refined sonic sketches and to think about sound in a more designerly way. This paper presents two main outcomes of the project: The Sound Design Toolkit, a palette of basic sound synthesis models grounded on ecological perception and physical description of sound-producing phenomena, and SkAT-Studio, a visual framework based on sound design workflows organized in stages of input, analysis, mapping, synthesis, and output. The integration of these two software packages provides an environment in which sound designers can go from concepts, through exploration and mocking-up, to prototyping in sonic interaction design, taking advantage of all the possibilities offered by vocal and gestural imitations in every step of the process.

1. INTRODUCTION

Sonic Interaction Design (SID) emerged in the last few years as a new area of design science, to overcome the lack of proper design attitude and process in the exploration of innovative uses of sound for interactive products, systems and environments [1]. Its research path has been moving from the understanding of sound perception, to the definition of sound modeling approaches for design, towards a progressive, deeper understanding of how sound designers think, how they learn to think in a designerly way, and how they develop their skills and knowledge [2, 3]. The discipline proposes a systematic approach for designing acoustic interactive behaviors by means of an iterative yet linear process, made of fixed and sequential steps which emphasize the importance of the conceptual phase, the fundamental value of the expressive qualities of sound in terms of character and identity, and the holistic view of sound creation in relation to the overall design of an artefact [4].

Investigation of the early stages of the sound design process is one of the most recent and promising research tracks in this context. Like in every other design activity, *sketching* is at the core of the initial conceptual phase. Sketches are quick, disposable and incomplete representations used to embody reasoning, communicate concepts, explore divergent ideas and eventually address the design process. In visual design, pencil and paper are still the most effective sketching tools, despite all technological advances. From architectural plans to page layouts, from paper models to graphical user interface mock-ups, drawings are extensively used throughout the design process to inform and support the progressive refinement of design ideas towards the final product [5].

In the aural domain, where a direct counterpart of pencil and paper is not available yet, a promising alternative is represented by *vocal sketching*. The practice exploits the human ability in the production of non-verbal utterances and gestures to imitate the main features of a given referent sound [6]. The human voice is extremely effective in conveying rhythmic information, whereas gestures are especially used to depict the textural aspects of a sound, and concurrent streams of sound events can be communicated by splitting them between gestures and voice [7]. Despite being embodied tools, immediately available to everyone [8] and increasingly popular in education and research [9, 10], the use of voice and gesture for sonic sketching is hardly spreading among sound practitioners, especially because of the inherent ephemerality of vocal/gestural representations and because of the limited timbral palette of the human voice.

A set of interviews with eight professional sound designers was conducted by the authors, to better understand the role of sketching in sound creation practices: The conceptual phase is mostly based on browsing sound banks and/or verbally describing concepts through a lists of keywords, while sonic sketching is still a neglected practice. Pressing time constraints and the lack of a shared language between designers and clients severely affect the search quality in the conceptual phase, resulting in conservative approaches and presentation of advanced design proposals even at the

very beginning of the process. When it is used, voice mostly serves as raw material for further sound processing and rarely as real-time control, while the use of gesture is limited to the operation of knobs and faders in musical interfaces. Finally, there is a pressing and unsatisfied demand for tools which are immediate to use, providing direct accessibility to sound production and design and facilitating the time consuming activity of finding a sensible mapping between control features and synthesis parameters.

The EU project SkAT-VG [1] (Sketching Audio Technologies using Vocalization and Gesture) aims at providing sound designers with a paper-and-pencil equivalent to seamlessly support the design process from the conceptual stage to prototyping. The goal is pursued through the development of computer-aided tools, using vocal and gestural imitations as input signals to appropriately select and control configurations of sound synthesis models according to the context of use [11, 12]. These tools aim at expanding the timbral possibilities of human sound production, while retaining the immediacy and intuitiveness of vocal articulation.

The use of voice to control the production of synthesized sound has already well established foundations in the musical domain. In his PhD thesis, Janer extracts audio descriptors from singing voice for the real-time control of pitch, volume and other timbral features in physical models of actual musical instruments such as bass, saxophone and violin [13]. Fasciani proposes an interface that allows to dynamically modify the synthesis timbre of arbitrary sound generators using dynamics in the vocal sound, exploiting machine learning techniques to perform the mapping between vocal audio descriptors and synthesis parameters [14]. Analysis of gestural features and their mapping for the control of digital musical instruments is also a widely explored domain [15].

These concepts can be translated from the context of musical performance to the field of Sonic Interaction Design. Our interest is focused on vocal and gestural production which is neither organized according to musical criteria nor in verbal form, and on sound synthesis techniques for the reproduction of everyday sounds and noises rather than digital musical instruments. Such a radically different context requires novel strategies in terms of analysis, mapping and synthesis. From now on, we will refer to our approach as *imitation-driven* sound synthesis, to differentiate it from previous related work focused on musical production.

The SkAT-VG project produced at least two relevant outcomes: the *Sound Design Toolkit* (SDT), a collection of sound synthesis algorithms grounded on ecological perception and physical description of sound-producing phenomena, and *SkAT-Studio*, a framework based on sound design workflows organized in stages of input, analysis, mapping, synthesis and output. Taken together, SDT and SkAT-Studio offer an integrated environment to go from the sonic sketch to the prototype: The input stage of SkAT-Studio accepts vocal and gestural signals, which are fed to the analysis stage to extract their salient features. This higher-level description of the input is then used by the mapping stage to control the synthesis stage, which embeds SDT modules and other sound synthesis engines.

The rest of the paper is organized as follows: The Sound Design Toolkit and its software architecture are described in Section 2; SkAT-Studio is covered in detail in Section 3; Section 4 explains how the two software packages can be integrated to achieve imitation-driven synthesis; Finally, conclusions and possible future work are exposed in Section 5.

2. THE SOUND DESIGN TOOLKIT

The Sound Design Toolkit is a collection of physically informed models for interactive sound synthesis, arranged in externals and patches for the *Cycling '74 Max* [2] visual programming environment. It can be considered as a virtual Foley box of sound synthesis algorithms, each representing a specific sound-producing event.

2.1 Conceptual framework

The development legacy of the SDT [2] dates back to the foundational research on the possibilities of interaction mediated by sound, and the importance of dynamic sound models in interfaces [16, 17]. Perceptual relevance has been a key concern in the selection and veridical reproduction of the acoustic phenomena simulated by the available sound models.

In his foundational work on the ecological approach to auditory event perception, Gaver proposed an intuitive hierarchical taxonomy of everyday sounds, based on the specific properties and temporal evolution of interacting materials [18]. In his taxonomy, the whole world of everyday sounds was described in terms of solids, liquids, gases interactions, their temporally-patterned evolution, and possible compounds. For example, the sound of writing was described by a compound deformation of impacts and patterned scraping events. Similarly, the sound of a motorboat was hypothesized as a high-level combination of gases, liquids, and solids interactions.

Originally based on Gaver's work [19], the SDT taxonomy of everyday sounds has been continuously revised, updated and extended over the years, to couple the sophistication of physically informed sound synthesis with the state of the art on the perception and categorization of environmental sounds [20, 21].

The design rationale behind the organization of the provided synthesis models is to encompass a mixture of sound categories, covering the major applications of sound design that are relevant for listeners, as shown in Figure 1. Sound models are grouped according to a criterion of causal similarity (i.e., vibrating solids, liquids, gasses, and machines) and arranged in a bottom-up hierarchy. The first level presents the basic algorithms with the corresponding Max externals, suitable for the generation of a large family of simple sound events. The second level highlights the basic processes and machines (with the corresponding Max externals), that can be either straightly derived from the temporal patterning of the low-level models or that would

[1] www.skatvg.eu.

[2] http://www.cycling74.com

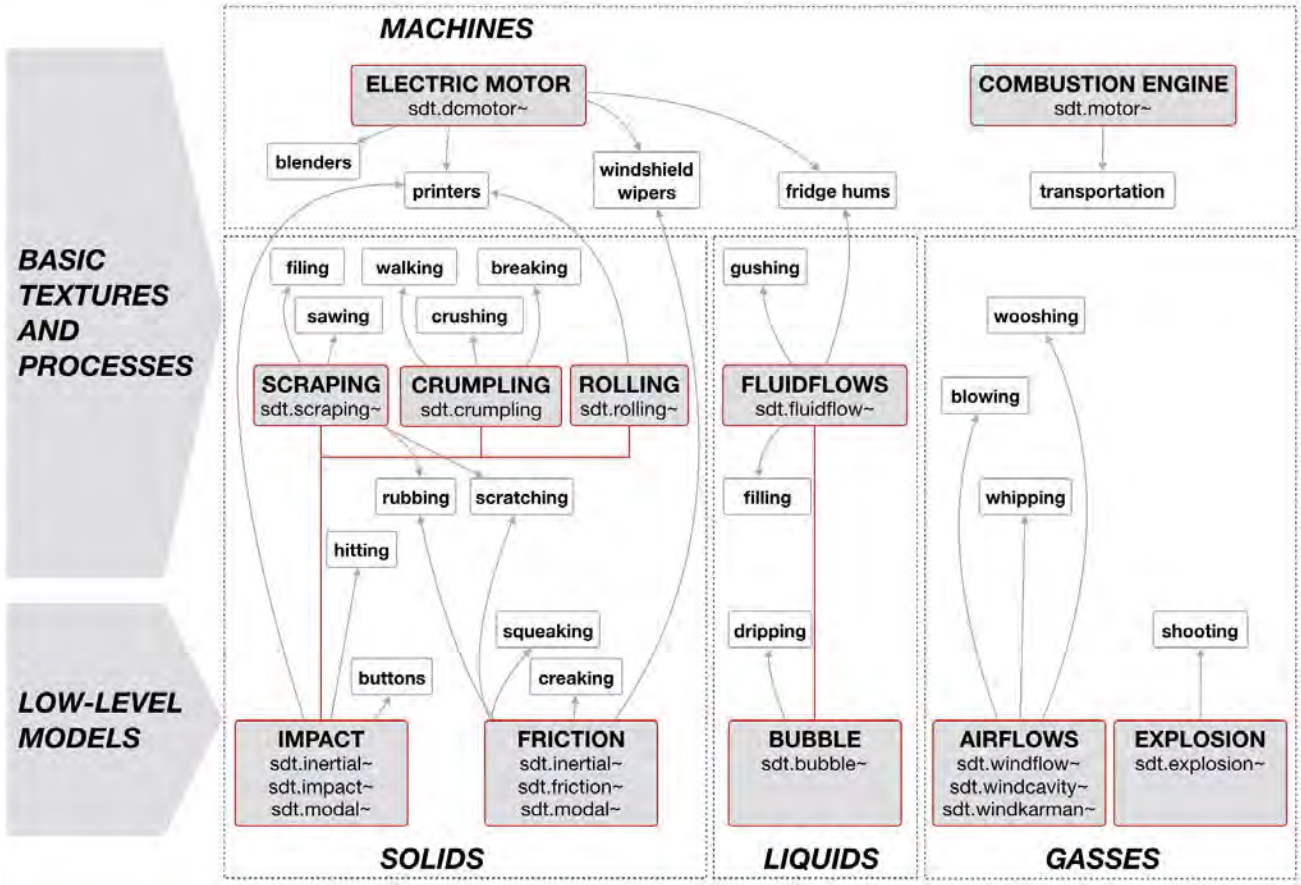


Figure 1. The SDT taxonomy of sound models. The bottom-up hierarchy represents the dependencies between low-level models and temporally-patterned textures and processes, for the four classes of sounds, solids, liquids, gasses, and machines.

be too cumbersome to develop as a Max chain of separate basic events.

In addition, the blue arrows set a direct connection between the sonic space of each model and the space of *timbral families*. Timbral families emerged from an extensive set of experiments on sound perception, as a higher-level classification of referent sounds that have been identified as cognitively stable in listeners' representations [22]. As seen from the SDT taxonomy viewpoint, a timbral family is defined as a peculiar parametrization of one or more sound synthesis models, which is unambiguously discriminated in terms of interaction, temporal and timbral properties.

2.2 Sound synthesis

The Sound Design Toolkit adopts a physically informed procedural approach to sound synthesis. In procedural audio, sound is synthesized from a computed description of the sound producing event, as opposed to sample-based techniques where sounds are prerecorded in a wavetable and then played back, manipulated and mixed together to obtain the desired timbral result [23]. Coherently with the conceptual foundation of the SDT, these computed descriptions are informed by the physics laws underlying the mechanical excitation and vibration involved in the sound events to reproduce.

The adoption of a simplified physics-based approach to sound modeling met the ecological and embodied instances emerging in computer-human interaction and design [24], thus grounding the development in design thinking and research [19]. Physically informed sound synthesis offers efficient, expressive and intuitive means to control and explore wide timbral spaces with a limited number of models, emphasizing the role of sound as a behavior, a process rather than a product. If it holds true that sound-producing events convey meaningful information about the underlying mechanical process, then manipulating their physical parameters should result in perceptually-relevant timbral modifications of the corresponding virtual sound.

The sound synthesis models are designed not only to be intuitively controllable by the user, but also to be computationally affordable for the machine. The desired efficiency is obtained through *cartoonification*, a specific design constraint implying a simplification of the physical descriptions and a consequent reduction of the available synthesis parameters. This economy of means exaggerates the most salient timbral aspects of the virtual sound events, a desired side effect which ultimately leads to a higher perceptual clarity of the simulation.

As previously mentioned in Section 2.1, the SDT sound models are used as basic building blocks to compose *timbral families*, categories of imitated sounds that are un-

ambiguously discriminable in terms of interaction, temporal and timbral properties. Whether composed by one or more low-level synthesis models, a timbral family is described in terms of specific, appropriate spaces and trajectories of sound synthesis parameters. All the timbral families (i.e., the blue boxes in Figure 1) are implemented and made available as Max patches in the current release of the toolkit.

2.3 A tool for sketching sonic interactions

Being temporary and disposable communication devices, sketches need to be produced with little time and effort. The more the resources required to produce a sketch, the greater the risk of being unwilling to throw it away in favor of possibly better options. The main advantage offered by drawn sketches in the early stages of a visual design process is the possibility to quickly materialize, store, compare and iteratively refine different ideas, gradually moving from early intuitions towards working prototypes.

The cartoonified, computationally affordable models of the SDT attempt to afford the same kind of interaction in the acoustic domain, enabling the sketching of sonic interactions in real-time on ordinary hardware, with a tight coupling between sound synthesis and physical objects to be sonified. The comparison and refinement of sonic sketches are made possible by means of saving and recalling presets of synthesis parameters. Presets can be further edited on GUIs or with MIDI/OSC external devices.

The almost direct relationship between synthesis parameters and basic physics facilitates understanding and creativity in sound design, supporting the unfolding of the designer's intentions on synthetic acoustic phenomena that are readily available and accessible through the concept of timbral family. Efforts are focused on providing economical control layers and parameter spaces, to interpret and control the physical descriptions of sound events in an intuitive way.

3. SKAT-STUDIO

SkAT-Studio is a prototype demonstration framework designed to facilitate the integration of other Max technologies in vocal and gestural sonic sketching.

3.1 Application workflow

A SkAT-Studio configuration is composed of the five following stages:

Input: Acquisition of voice and gesture;

Analysis: Extraction of meaningful features and descriptors from the input;

Mapping: Transformation of the analysis features into synthesis parameters by further elaboration, rescaling and/or combination;

Synthesis: Production of sound. This can be either purely procedural sound synthesis or post-processing of an existing sound (e.g. pitch shifting or time stretching);

Output: Playback or recording of the final sound.

3.2 Software overview

The framework is designed with flexibility and modularity in mind, and it is entirely developed in Max. It is composed by a main GUI (see Figure 2) which can host and link together a collection of loadable *modules*, each one taking care of a specific operation in the global process. Several modules can be loaded simultaneously, and signal and/or control data can be routed at will among different modules using *patchbays*.

Many different modules can be loaded at any given time, leading to possible cluttering of the interface and computing performance issues. To mitigate this problems, each of the five stages (input, analysis, mapping, synthesis, output) is materialized as a *group*. Groups help organizing information and simplifying the use of the software. Each group may contain several modules, whose control data and signals can be routed to other modules in the same group or even to an external group. Each SkAT-Studio module belongs to a group, according to its function.

A wide variety of modules is already available in the framework, offering the basic building blocks for the composition of complex configurations. The acquisition of audio signals from a microphone (input), the extraction of one feature or a set of features (analysis), the linear transformation of a parameter (mapping), the implementation of sound models (synthesis) and the direct playback through the speakers (output) are just some of the functions offered by the SkAT-Studio core modules.

In addition, users can easily build and add their own modules inside the SkAT-Studio framework. Each module is realized as a separate Max patch, which must adhere to a simple module template. The template provides a common interface for back-end communication with the other parts of the framework and front-end integration into the main GUI. To comply with the template, modules must graphically fit a given area, and provide the following information:

- Name of the module,
- Inputs and outputs of the module (number and names),
- Documentation (input/output data types, author, description of the underlying algorithms and so on).

The interactive and visual nature of the Max patching environment, combined with the simple yet versatile module template, allows quick and easy integration of new features into the system.

Audio signals and control data can be freely routed from any output of a module to any input of any other module, using routing matrices called *patchbays*. A patchbay is a double entry table, as displayed in Figure 3, with all the module outputs listed on the top row and all the module inputs listed on the left column. A toggle matrix allows to associate each output to one or more inputs, simply activating the appropriate toggles in the double entry table.

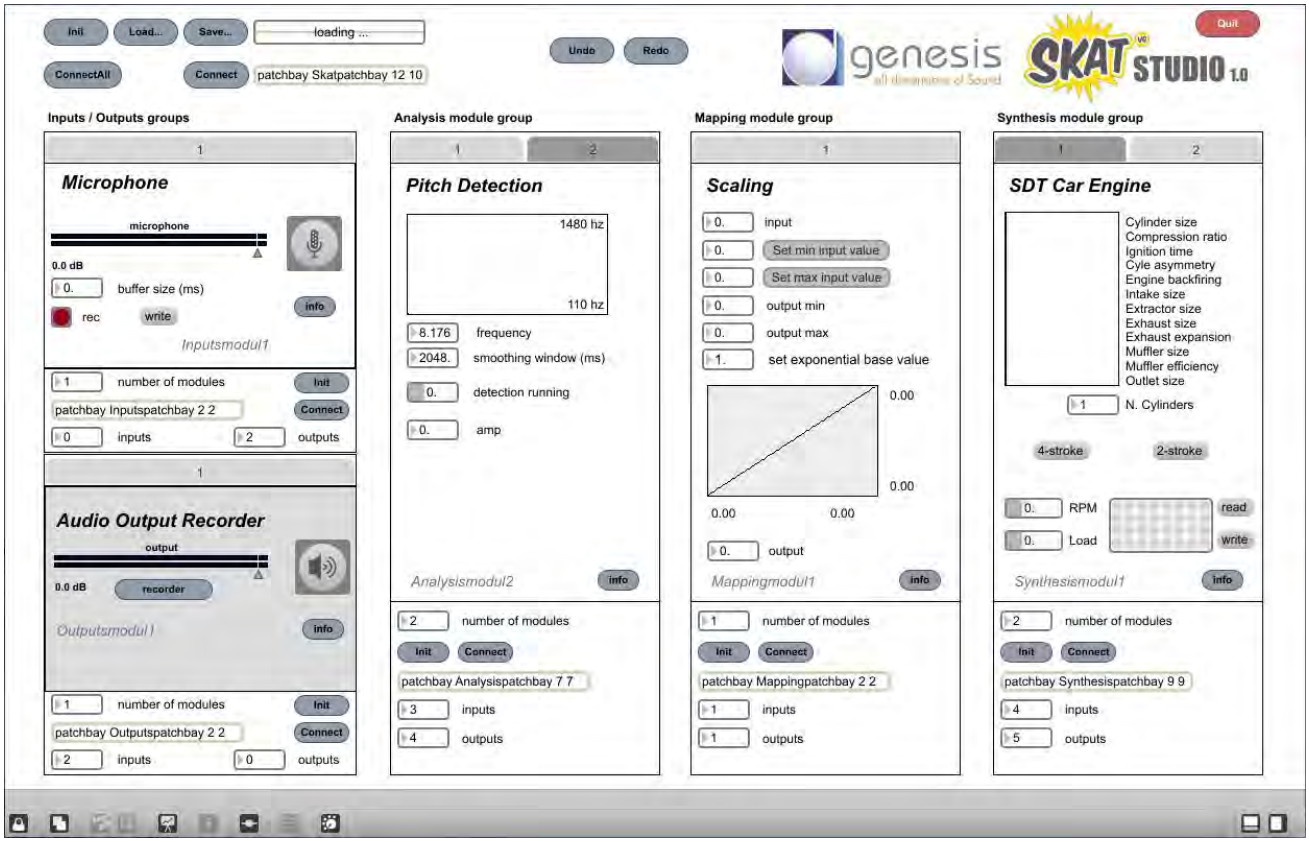


Figure 2. The SkAT-Studio workflow.

The subdivision of modules into functional groups, originally introduced to reduce conceptual and interface clutter, also simplifies the data routing process. Inbound and outbound data are first routed among modules inside a group, and successively among the groups inside the main framework. The framework therefore includes six patchbays: One for each group, plus a global one for the whole system.

3.3 Building a configuration

SkAT-Studio configurations can be built by performing a series of simple operations through the application GUI. The first step is choosing how many modules need to be loaded in each group. This operation creates as many tabs as required in the corresponding canvases. Modules can then be loaded in the tabs, either by drag and drop from a file manager or by choosing the module from the list visualized in the empty tab.

The next step is defining the number of inputs and outputs that each group should expose to the global routing patchbay of the system. By default, it is the total number of inputs/outputs of all the modules instantiated in the group. However, avoiding to expose data which do not need to go outside of the group allows to reduce the amount of routing connections, and therefore the size of the global patchbay.

Once everything is set up in place, the last step consists in clicking on the *connect* buttons to open the patchbays and route data inside of each group and among different groups. Once the configuration is ready, the sound designer can work with it and produce sonic sketches by tweaking the parameters exposed by the different modules. The possibility to save and load timbral family presets, together with an undo/redo history function, allows to compare, refine and possibly merge different sketches.

4. IMITATION-DRIVEN SOUND SYNTHESIS

The expressive power of human voice and gestures can be exploited to control the sound synthesis process and leveraged to perform quick and rough explorations of the parameters space of the available algorithms, shaping sound by mimicking the desired result. Taken together, the Sound Design Toolkit and SkAT-Studio provide an integrated environment for imitation-driven sound synthesis, in which sound designers can go from concepts, through exploration and mocking-up, to prototyping in sonic interaction design, taking advantage of all the possibilities offered by vocal and gestural imitations in every step of the process. The global workflow of the system is composed of two steps:

Select: The user produces a vocal imitation of the desired sound. The vocal imitation is recognized, classified, and the corresponding timbral family and vocal/gestural control layer are selected.

Play: The user controls the synthesizers in real time using vocalization and gesture, navigating the timbral space of the selected model and iteratively refining her sonic sketches. The use of voice and gesture to

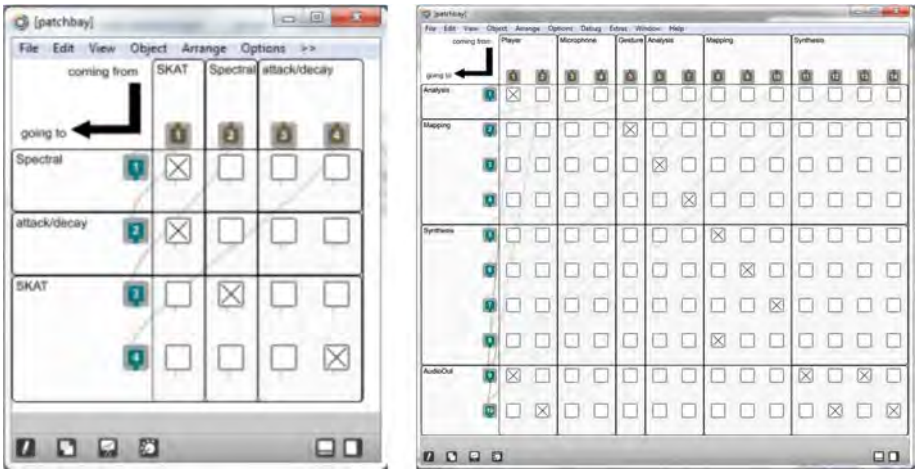


Figure 3. On the left, example of a patchbay for the analysis group. On the right, the global SkAT-Studio patchbay.

control sound production allows a fast, direct and easy manipulation of the synthesis parameters.

The *Select* step accepts a sound signal as input, and outputs a SkAT-Studio configuration which defines the behavior of the *Play* step.

The first step towards imitation-driven sound synthesis is the extraction of meaningful information from the vocal signal, in the form of higher level features and descriptors. To accomplish this task, the SDT has been enriched with tools for the analysis of audio signals in addition to the collection of sound synthesizers. A wide range of well-documented audio descriptors [25–27], have been reimplemented and made available as SDT externals. Recent studies on vocal imitations of basic auditory features and identification of sound events, however, pointed out that effective imitation strategies for the communication of sonic concepts exploit a few and simple acoustic features, and that the features cannot be consistently and reliably controlled all together, at the same time [28, 29].

In this respect, only a limited amount of descriptors is actually useful, and an even smaller subset is used to control a timbral family at any given time. Voice and gesture are used for a coarse control of the synthesis models, leaving further timbral refinement to manual operation on the graphical user interface or other external devices [12]. When placed on the visual canvas of the Max patcher, and connected in a coherent data flow, the SDT components can be operated via GUI sliders and knobs or external devices to refine the result. The extraction of features for control purposes includes:

- Amplitude variations and temporal patterns;
- Fundamental frequency, closely related to the sensation of pitch;
- Signal zero crossing rate, a rough estimate of the noisiness of a sound;
- Spectral centroid, directly related to the sensation of brightness of a sound;
- Spectral energy distribution, changing for different vowels.

Each of the SDT analysis externals is embedded in a separate SkAT-Studio module, to allow its inclusion in SkAT-Studio configurations.

The descriptors obtained by the analysis modules must then be mapped to the synthesis parameters of the available models and used to control the temporal behavior of the sound models. For each timbral family, a small subset of the available descriptors is scaled, combined and assigned to the vocally controlled synthesis parameters. All the operations involved in this process are performed by SkAT-Studio modules belonging to the mapping group.

At this stage, a simple, yet effective set of control maps per timbral family has been devised, which meets the listener expectations about the behavior of the sound producing events. For example, as the energy of an impact is expected to affect the amplitude and the spectral bandwidth of the resulting sound, similarly the timbral characteristics of its imitation will produce the same effect. In other words, it is possible to exploit the common relations between timbral features and physical parameters. Some examples include:

- The *pitch* of a vocal signal can be directly mapped to the revolutions per minute of both combustion engines and electric motors;
- The spectral *centroid* can be related to the concept of size (for instance, the size of bubbles in liquid sounds);
- The spectral *spread* can be associated to the concept of hollow body resonance, as found in many timbral families (e.g cavities in an air flow, the chassis of an electric motor, the exhaust system of a combustion engine, a container filled with a liquid, etc.);
- The temporal and spectral *onset* information can be used to trigger discrete events, like single impacts or explosions;

- The *zero crossing rate* of a vocal imitation can be put in relation with the graininess in higher level textures such as rolling, rubbing, scraping and crumpling, to the harshness of machine sounds, and in general to all the synthesis parameters related to the concept of noisiness.

Finally, the output of the mapping modules is routed to the synthesis group, to generate the sonic sketch. Each timbral family defined in the SDT is ported into SkAT-Studio as a synthesis module, exposing the vocally controlled synthesis parameters as inputs and the generated audio signal as output. Although not all the timbral possibilities provided by the synthesis modules are reproducible and controllable by vocal imitations, it is nevertheless possible to produce convincing and recognizable sonic sketches by mimicking a few salient, perceptually-relevant features for their identification. More subtle nuances, not directly controllable by vocal input, can be tweaked on the GUI of each module using traditional input methods such as virtual sliders and knobs.

To summarize, the proposed framework strives to facilitate the sound designer by providing models of sounds that humans can think of and represent through their voice and gestures. This aspect is reflected in the general procedural audio approach informing the SDT algorithms, and in the organization of SkAT-Studio workflows and configurations.

5. CONCLUSIONS AND FUTURE WORK

Although not fully evaluated yet, SDT and SkAT-Studio have been successfully used together for sketching the sonic behavior of a driving simulator, in the context of virtual reality and augmented environments [30]. Imitation-driven sound synthesis has also been presented and used in a series of sound design workshops, conducted as part of the SkAT-VG project.

We recently involved expert sound designers, in the 48 Hours Sound Design workshop [3] at Chateau La Coste art park and vineyard, in south France. Five professional sound designers were invited to work each on one of the site-specific art pieces located in the park, and design an accompanying sound signature for the chosen art installation, in 48 hours. Vocal sketching methods and tools (including SkAT Studio and SDT) were the exclusive means available for sound ideas generation and sketching. In general, the technological support to vocal production and sketching was positively received, as the sound designers managed to explore and produce a large set of sounds in a very limited set of time. Yet, the provided SDT palette of sound models was found to be too bounded to realistic behaviors. The sound designers were also concerned about the cartoonified quality of the resulting sound. However, this rather reflected their inclination to produce well-refined sound propositions from the very beginning of their creative process, thus stressing a certain reluctance towards sketching and its purpose.

[3] The documentary of the workshop is available at: `https://vimeo.com/169521601`.

Indeed, vocal sketching in cooperative sound design tasks have been extensively documented, during a recent workshop held in November 2015 at the Medialogy course of Aalborg University Copenhagen, Denmark, and it is currently undergoing a process of detailed protocol and linkographic analyses [31]. Protocol and linkographic analyses are aimed at producing a fine-grained understanding of the cognitive behaviors in sound design tasks, measure the efficiency of the creative process, and ultimately assess the effectiveness of vocal sketching methods. Hence, the design of the sketching tools is grounded in the development of skills and practices of sound representations.

Acknowledgments

The authors are pursuing this research as part of the project SkAT-VG and acknowledge the financial support of the Future and Emerging Technologies (FET) programme within the Seventh Framework Programme for Research of the European Commission, under FET-Open grant number: 618067.

6. REFERENCES

[1] K. Franinović and S. Serafin, *Sonic interaction design: fresh perspectives*. Mit Press, 2013.

[2] D. Rocchesso, "Sounding objects in europe," *The New Soundtrack*, vol. 4, no. 2, pp. 157–164, 2014.

[3] S. Delle Monache and D. Rocchesso, "Bauhaus legacy in research through design: The case of basic sonic interaction design," *International Journal of Design*, vol. 8, no. 3, pp. 139–154, 2014.

[4] D. Hug and N. Misdariis, "Towards a conceptual framework to integrate designerly and scientific sound design methods," in *Proceedings of the 6th Audio Mostly Conference: A Conference on Interaction with Sound*, ser. AM '11. New York, NY, USA: ACM, 2011, pp. 23–30. [Online]. Available: http://doi.acm.org/10.1145/2095667.2095671

[5] S. Greenberg, S. Carpendale, N. Marquardt, and B. Buxton, *Sketching user experiences: The workbook*. Boston: Morgan Kaufmann, 2012.

[6] G. Lemaitre and D. Rocchesso, "On the effectiveness of vocal imitations and verbal descriptions of sounds," *The Journal of the Acoustical Society of America*, vol. 135, no. 2, pp. 862–873, 2014.

[7] H. Scurto, G. Lemaitre, J. Françoise, F. Voisin, F. Bevilacqua, and P. Susini, "Combining gestures and vocalizations to imitate sounds," *The Journal of the Acoustical Society of America*, vol. 138, no. 3, pp. 1780–1780, 2015.

[8] I. Ekman and M. Rinott, "Using vocal sketching for designing sonic interactions," in *Proceedings of the 8th ACM Conference on Designing Interactive Systems*. ACM, 2010, pp. 123–131.

[9] B. Caramiaux, A. Altavilla, S. G. Pobiner, and A. Tanaka, "Form follows sound: Designing interactions from sonic memories," in *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*, ser. CHI '15. New York, NY, USA: ACM, 2015, pp. 3943–3952. [Online]. Available: http://doi.acm.org/10.1145/2702123.2702515

[10] C. Erkut, S. Serafin, M. Hoby, and J. Sårde, "Product sound design: Form, function, and experience," in *Proceedings of the Audio Mostly 2015 on Interaction With Sound*, ser. AM '15. New York, NY, USA: ACM, 2015, pp. 10:1–10:6. [Online]. Available: http://doi.acm.org/10.1145/2814895.2814920

[11] D. Rocchesso, G. Lemaitre, P. Susini, S. Ternström, and P. Boussard, "Sketching sound with voice and gesture," *Interactions*, vol. 22, no. 1, pp. 38–41, 2015.

[12] D. Rocchesso, D. Mauro, and S. Delle Monache, "miMic: the microphone as a pencil," in *Proceedings of the Tenth International Conference on Tangible, Embedded and Embodied Interaction*, ser. TEI '16. ACM, 2016.

[13] J. Janer, "Singing-driven interfaces for sound synthesizers," Ph.D. dissertation, Universitat Pompeu Fabra, Barcelona, 2008.

[14] S. Fasciani and L. Wyse, "A voice interface for sound generators: adaptive and automatic mapping of gestures to sound." in *Proceedings of the 12th International Conference on New Interfaces for Musical Expression (NIME)*, 2012.

[15] M. M. Wanderley and P. Depalle, "Gestural control of sound synthesis," *Proceedings of the IEEE*, vol. 92, no. 4, pp. 632–644, 2004.

[16] W. W. Gaver, "The sonicfinder: An interface that uses auditory icons," *Hum.-Comput. Interact.*, vol. 4, no. 1, pp. 67–94, Mar. 1989. [Online]. Available: http://dx.doi.org/10.1207/s15327051hci0401_3

[17] M. Rath and D. Rocchesso, "Continuous sonic feedback from a rolling ball," *IEEE MultiMedia*, vol. 12, no. 2, pp. 60–69, 2005.

[18] W. W. Gaver, "What in the world do we hear?: An ecological approach to auditory event perception," *Ecological psychology*, vol. 5, no. 1, pp. 1–29, 1993.

[19] S. Delle Monache, P. Polotti, and D. Rocchesso, "A toolkit for explorations in sonic interaction design," in *Proceedings of the 5th Audio Mostly Conference: A Conference on Interaction with Sound*, ser. AM '10. New York, NY, USA: ACM, 2010, pp. 1:1–1:7. [Online]. Available: http://doi.acm.org/10.1145/1859799.1859800

[20] G. Lemaitre, O. Houix, N. Misdariis, and P. Susini, "Listener expertise and sound identification influence the categorization of environmental sounds." *Journal of Experimental Psychology: Applied*, vol. 16, no. 1, p. 16, 2010.

[21] O. Houix, G. Lemaitre, N. Misdariis, P. Susini, and I. Urdapilleta, "A lexical analysis of environmental sound categories." *Journal of Experimental Psychology: Applied*, vol. 18, no. 1, p. 52, 2012.

[22] G. Lemaitre, F. Voisin, H. Scurto, O. Houix, P. Susini, N. Misdariis, and F. Bevilacqua, "A large set of vocal and gestural imitations," SkAT-VG Project, Tech. Rep., November 2015, deliverable D4.4.1. [Online]. Available: http://skatvg.iuav.it/wp-content/uploads/2015/11/SkATVGDeliverableD4.4.1.pdf

[23] A. Farnell, "Behaviour, structure and causality in procedural audio," in *Game sound technology and player interaction concepts and developments*, M. Grimshaw, Ed. New York, NY, USA: Information Science Reference, 2010, pp. 313–329.

[24] D. Svanæs, "Understanding interactivity: Steps to a phenomenology of human-computer interaction, monograph," Ph.D. dissertation, NTNU, Trondheim, Norway, 2000. [Online]. Available: http://dag.idi.ntnu.no/interactivity.pdf

[25] G. Peeters, B. L. Giordano, P. Susini, N. Misdariis, and S. McAdams, "The timbre toolbox: Extracting audio descriptors from musical signals," *Journal of the Acoustical Society of America*, vol. 130, no. 5, p. 2902, 2011.

[26] P. McLeod and G. Wyvill, "A smarter way to find pitch," in *Proceedings of International Computer Music Conference, ICMC*, 2005.

[27] D. Stowell and M. Plumbley, "Adaptive whitening for improved real-time audio onset detection," in *Proceedings of the International Computer Music Conference (ICMC 07)*, Copenhagen, Denmark, 2007.

[28] G. Lemaitre, A. Dessein, P. Susini, and K. Aura, "Vocal imitations and the identification of sound events." *Ecological psychology*, vol. 23, no. 4, pp. 267–307, 2011.

[29] G. Lemaitre, A. Jabbari, O. Houix, N. Misdariis, and P. Susini, "Vocal imitations of basic auditory features," *The Journal of the Acoustical Society of America*, vol. 137, no. 4, pp. 2268–2268, 2015.

[30] S. Baldan, H. Lachambre, S. D. Monache, and P. Boussard, "Physically informed car engine sound synthesis for virtual and augmented environments," in *Proceedings of the 2nd VR Workshop on Sonic Interactions for Virtual Environments (SIVE)*. IEEE, 2015.

[31] S. Delle Monache and D. Rocchesso, "Cooperative sound design: A protocol analysis," in *Accepted for publication in Proc. of AudioMostly 2016, a conference on interaction with sound*, Norrköping, Sweden, 2016.

SEED: RESYNTHESIZING ENVIRONMENTAL SOUNDS FROM EXAMPLES

Gilberto Bernardes
INESC TEC
gba@inesctec.pt

Luis Aly
University of Porto, Faculty of Engineering
luis.aly@fe.up.pt

Matthew E. P. Davies
INESC TEC
mdavies@inesctec.pt

ABSTRACT

In this paper we present SEED, a generative system capable of arbitrarily extending recorded environmental sounds while preserving their inherent structure. The system architecture is grounded in concepts from concatenative sound synthesis and includes three top-level modules for segmentation, analysis, and generation. An input audio signal is first temporally segmented into a collection of audio segments, which are then reduced into a dictionary of audio classes by means of an agglomerative clustering algorithm. This representation, together with a concatenation cost between audio segment boundaries, is finally used to generate sequences of audio segments with arbitrarily long duration. The system output can be varied in the generation process by the simple and yet effective parametric control over the creation of the natural, temporally coherent, and varied audio renderings of environmental sounds.

1. INTRODUCTION

The need to extend a given environmental audio sample is a recurrent problem in sound design [1, pp. 38-39]. A typical example in sound post-production for television and film is when pre-recorded audio does not cover the entire duration of a scene. The most common solution is to manually find a smooth transition point in the sample and loop it [2, pp. 178, 204]. Since these tracks typically run in the background, this solution is acceptable despite being time-consuming and recognizably repetitive. A different scenario in which the need to extend a given environmental audio sample is when the file size of the audio content is an important consideration. This concerns the memory storage and/or the time needed to download or stream media assets in applications such as videogames, installations, and screensavers. Mostly, these applications use fixed audio samples, which means that greater variation implies more storage. Guaranteeing a small memory storage footprint is critical to the success of such applications.

A possible solution is to extend the duration of a given short environmental sound recording arbitrarily. While, to the best of our knowledge, no commercial applications exist to fulfill this purpose, academic research has recently looked at this problem [3-7]. For a comprehensive review of system for the expansion of environmental sounds refer to [8, 9]. Conducted research in this topic relies on the fact that, within a particular time and geographical span, environmental sounds are i) relatively monotonic, ii) simple in structure, and iii) have a high degree of redundancy [8]. Existing systems typically follow a threefold structure that handles segmentation, analysis and the resynthesis of a given environmental sound example.

Current technological solutions are mostly confined to the extension of simple sound textures and barely address environmental sounds with complex temporal structures. Additionally, most systems extend a given input file by generating new sequences based on the similarity among audio segments in order to create smooth transitions [4, 5]. While this approach provides good results for highly redundant audio content, it does not provide an optimal answer to the problem, especially when processing environmental sound scenes with a higher degree of complexity and temporal dependencies, such as moving vehicles, and storms.

Our work strives for a solution capable of generating long audio streams from a short input audio signal example of non-diegetic sounds excluding dialogue with variable complexity by capturing its intrinsic structure. Based on representations of an input environmental sound we propose a real-time system that generates new audio sequences of arbitrary duration with dynamic control over the amount of novelty introduced. We expand on previous research with novel methods for audio segmentation and learning audio structures based on a flexible dictionary-based representation of audio classes derived from clustering methods. Furthermore, we use a concatenation cost—a concept borrowed from concatenative sound synthesis and in particular from [10]—to measure the transition between audio segments towards generating smooth audio sequence transitions.

A prototype application, named SEED (Sound Environmental ExpanDer), implements the model detailed here in Pure Data.[1] SEED is composed of three top-level modules, detailed in the following three sections, responsible for i) segmenting an environmental input audio signal (Section 2), ii) creating a temporal model of the input signal (Section 3), and iii) generating new arbitrarily-long audio streams based on the audio input structure (Section 4). The last three sections present SEED's graphical user interface (Section 5), an evaluation of the system (Section 6), conclusions, and directions for future work (Section 7).

[1] https://puredata.info/, last access on January 2016.

2. AUDIO INPUT SEGMENTATION

In SEED, we adopt an audio segmentation strategy, which isolates events in time with clear spectral differences. Inspired by the work of Hoskinson and Pai [4], the boundaries of each segment correspond to stable moments (lowest spectral difference between audio analysis frames), aiming to favor smoother transitions between resynthesized audio segments during generation.

In greater detail, we first compute the spectral flux function $SF(m)$ of non-normalized audio frames m (window size ≈ 46.4 ms and window overlap of 50%) using the timbreID library [11]. Then, in order to minimize spurious detections, we smooth the spectral flux function $SF(m)$ by a bidirectional (or zero-phase shift) first-order infinite impulse response low-pass filter $\widehat{SF}(m)$ with cutoff frequency of 5 Hz.

The third step of the algorithmic chain is a valley-picking algorithm that defines segment boundaries. Valleys are computed by finding all local (or relative) minima on the filtered spectral flux function $\widehat{SF}(m)$ below a dynamic threshold value $t(m)$. The threshold $t(m)$ aims to regulate magnitude changes across the temporal dimension of the filtered spectral flux function $\widehat{SF}(m)$. It is computed as the difference between the local median $\mu(m)$ and the local standard deviation $\sigma(m)$ of a window of size 32 analysis frames around m in the filtered spectral flux function $\widehat{SF}(m)$ such that:

$$t(m) = \mu(m) - \alpha \cdot \sigma(m) \quad (1)$$

where α bias the relative weight of the standard deviation and is set to $\alpha = 0.5$.

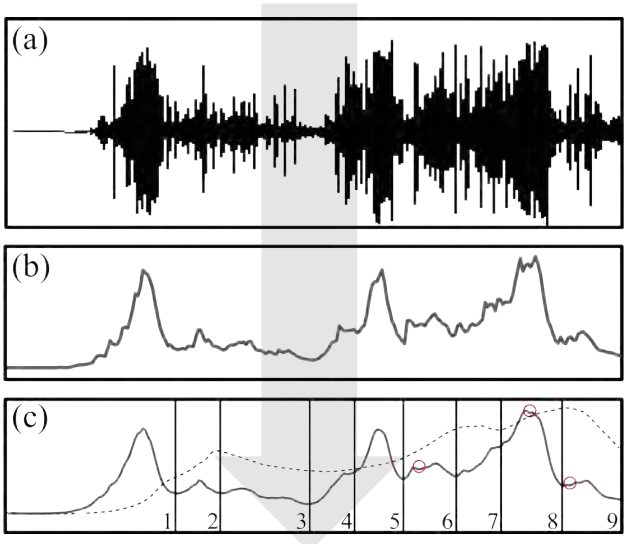


Figure 1. Illustration of the audio input segmentation in three stages: (a) waveform, (b) spectral flux function $SF(m)$, and (c) filtered spectral flux function $\widehat{SF}(m)$ on which we apply a valley picking algorithm which defines segment boundaries—circles indicate valleys discarded by $t(m)$, represented as a dashed line, and the local proximity constraint.

We additionally discard valleys that are fewer than four analysis frames apart to guarantee segments with a minimum duration of 185.8 ms wherein we apply a cross fade during generation. In Figure 1, from the three-circled valleys, the middle was excluded because it was greater than $t(m)$, and the two others were excluded because their position was fewer than four analysis frames apart from a previously detected valley.

Finally, in order to guarantee a minimum number of segments in case the input audio signal is purely composed of highly stationary sounds (or highly stable spectra), the system arbitrarily segments the input audio signal until a ratio r of the total duration of the signal (in seconds) to the total number of segments S reaches a number below unity.

3. LEARNING OPTIMAL TRANSITIONS

In view of our goal to generate environmental audio that preserves the inherent structure of an input signal, we create two representations of its structure. The first is detailed in Section 3.1 and consists of a table encoding transitions between *classes* representing the input audio segments. Prior to the table creation, segments are represented by a set of features extracted from the signal content with higher degree of variability (which due to our segmentation strategy is most likely to be in the middle region of a segment) and then clustered into classes. The second is detailed in Section 3.2 and consists of matrix encoding the concatenation cost among all possible segment transitions computed by comparing features from the troughs of the audio input segments.

3.1 Encoding the Audio Input Temporal Dynamics

Transition tables are a commonly applied strategy to encode finite and discrete high-level symbolic music patterns for style imitation [12]. They are easy to implement, computationally efficient and, when modeling structures with enough redundancy, have proven to be powerful in generating similar musical sequences to the structure they represent, while ensuring variation [12].

When derived from audio, transition tables are particularly difficult to compute, due to the low-level and noisy representation of the signal. The biggest challenge is to parameterize the audio using a finite and discrete space, while capturing its multidimensional attributes. To this end, we propose a two-step parameterization process, detailed in Sections 3.3.1 and 3.3.2, which groups clusters (pre-segmented) audio events into classes. The generated audio classes are then used to compute a transition table, which provides the basis for the generative process.

3.1.1 Parameterizing the Audio Signal

The first audio parameterization step represents each audio segment by the following collection of five audio features: spectral brightness, spectral flatness, zero crossing rate, spectral spread, and amplitude. These features are among the audio descriptors from Brent's timbreID library [11] chosen on the basis of their relevancy to describe the spectro-temporal dimension of environmental sound sources [13, 14].

For each segment, we first extract the abovementioned audio features on an overlapping window basis (window size ≈ 11.6 ms, and window overlap of 50%). Then, for each segment, we compute first and second order statistics (i.e., min, max, mean and variance). A feature vector of 20 (5 x 4) elements per audio segment is finally stored in a database.

To enhance the audio descriptions, we adopt an automatic strategy to weight each descriptor according to its variance across the entire input audio signal. This strategy, first proposed in [11] and explored in the realm of audio generative contexts in [15], assumes that features with higher variance across an audio example are more relevant, because their high variance provides a more distinctive characterization of the segments. Prior to the weight's assignment, we normalize each descriptor to the 0-1 range by their minimum and maximum range values.

3.1.2 A Dictionary of Audio Events

As discussed in Section 3.1, a major difficulty to represent audio segments in transition tables is to reduce them to a finite and discrete set of representations that capture their multidimensional audio content attributes. In Section 3.1.1, we already showed a strategy to represent an audio segment as a 20-element feature vector. Now, we perform additional data reduction by creating a dictionary of sound classes that represent the sound segments by a unique value.

Segment class creation is performed by an agglomerative hierarchical clustering algorithm [16], a method inspired by the work of Saint-Arnaud and Popat [17] on sound texture analysis and synthesis, which we extend by proposing several clustering solutions with variable numbers of elements (or audio classes) for a given audio input source. Our aim behind the choice of this particular algorithm was to allow a user to drive or adapt the dictionary construction intuitively by choosing the number of audio classes it includes. In doing so, we additionally avoid the formalization of many subjective and contextual factors inherent to the task (please refer to [15, pp. 74-79] for a broader discussion on this topic).

In agglomerative hierarchical clustering, each of the S input audio segments starts as its own cluster, and iteratively the algorithm pairs them until it reaches a configuration where all S segments belong to one cluster. In order to decide which audio segments are paired at each iteration, a similarity metric and a linkage criterion—which specifies how pairwise distances involving clusters with more than one segment are computed—is required. In our work, the Euclidean distance between audio segment feature vectors expresses similarity (small distances correspond to similar segments and large distance to dissimilar ones). Clustered segments are represented by the mean vector values of their constituent segment feature vectors. Consequently, inter-cluster distances are computed as the Euclidean distance between two such mean vectors—referred to as the centroid method [16].

Hierarchical clustering is commonly illustrated by a dendrogram, i.e. a tree structure that displays distances amongst input data elements and clusters (see Figure 2). Horizontal lines connect the most similar elements at a given iteration, thus forming a "new" element. The distance of a particular pair of segments or segment clusters is reflected in the height of the horizontal line.

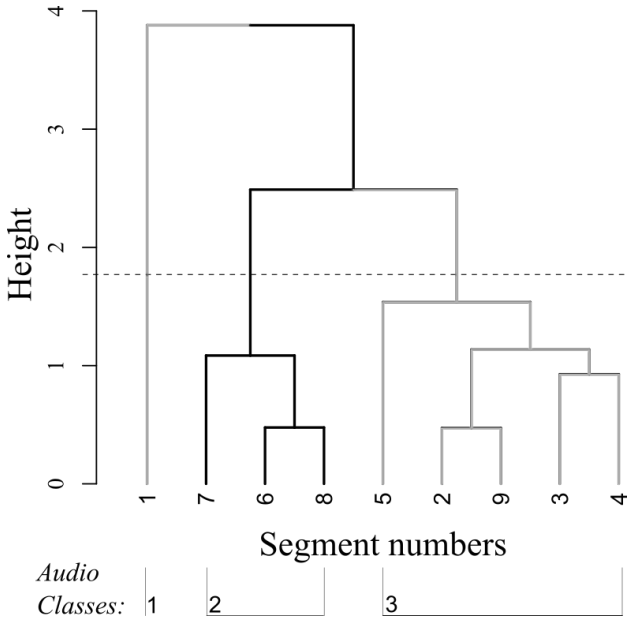


Figure 2. Dendrogram illustrating the hierarchical clustering of the audio segments shown in Figure 1. The dashed line shows the cutting point that instantiates three clusters expressed by alternating colors.

Table 1 shows all possible cluster configurations from the audio segments shown in Figure 1 as illustrated in the dendrogram in Figure 2. All possible numbers of clusters from nine clusters to a single cluster are illustrated. A hierarchy from the resulting set of clusters needs to be selected by user. This is equivalent to cutting the dendrogram at a particular height. The cutting point in Figure 2 corresponds to the configuration with three clusters. For a given solution, we then assign labels to each cluster using an increasing sequence of natural numbers to build a dictionary of audio classes (shown in Figure 2 and in the third column of Table 1 for all cluster hierarchies).

The higher the hierarchy of the agglomerative clustering algorithm, the greater redundancy in the creation of the audio classes dictionary.

Number of clusters	Clusters	Audio classes
9	1, 2, 3, 4, 5, 6, 7, 8, 9	1, 2, 3, 4, 5, 6, 7, 8, 9
8	1, {2, 9}, 3, 4, 5, 6, 7, 8	1, 2, 3, 4, 5, 6, 7, 8
7	1, {2, 9}, 3, 4, 5, {6, 8}, 7	1, 2, 3, 4, 5, 6, 7
6	1, {2, 9}, {3, 4}, 5, {6, 8}, 7	1, 2, 3, 4, 5, 6
5	1, {2, 9}, {3, 4}, 5, {6, 7, 8}	1, 2, 3, 4, 5
4	1, {2, 3, 4, 9}, 5, {6, 7, 8}	1, 2, 3, 4
3	**1, {2, 3, 4, 5, 9}, {6, 7, 8}**	**1, 2, 3**
2	1, {2, 3, 4, 5, 6, 7, 8, 9}	1, 2
1	{1, 2, 3, 4, 5, 6, 7, 8, 9}	1

Table 1. Cluster configurations from the audio segments shown in Figure 1 as illustrated in the dendrogram in Figure 2. Sets within brackets denote clustered audio segments.

3.1.3 Audio Classes Transition Table

To encode the structure of the input audio signal, we finally create a table representing its component audio classes transitions. We first create a string of numbers representing the temporal dimension of the input audio signal by substituting each segment number by its representative audio class (see Figure 3). A database establishing the correspondence between audio classes and their component audio segments is then created. Following the example shown in Figure 3, our database would include the following entries: {**1**: 1}; {**2**: 2 3 4 5 9}; and {**3**: 6 7 8}. The first index element is the audio class number and the following values (after the comma) correspond to the audio segments of that class.

Finally, we build a table encoding all possible transitions between audio classes. Based on the example shown in Figure 3, for a first-order transition table, we would have the sequences shown in Figure 4.

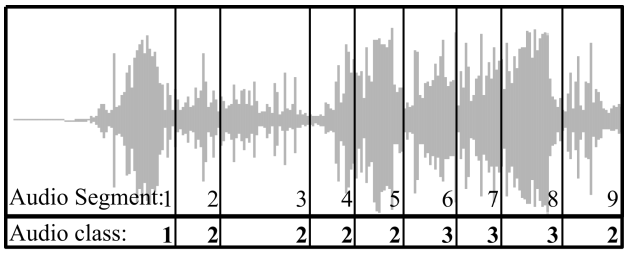


Figure 3. Correspondence between audio segments and audio classes derived from the hierarchical clustering tree shown in Figure 2.

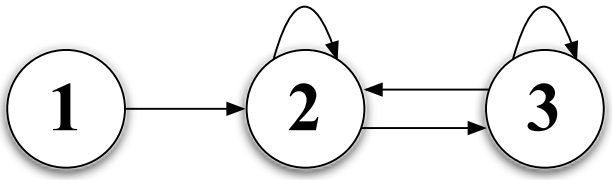


Figure 4. Visualization of all possible transitions among the audio classes from the example shown in Figure 3.

The order of the transition table is by default assigned to a third-order (chosen on the grounds of heuristic evaluations of various audio input signals with durations ranging from 30 seconds to a minute long) and can be manually changed by the user.

3.2 Audio Segments Concatenation Cost

An S x S matrix C expressing the concatenation cost between all audio segments is created to promote smooth transitions during generation. Given two segments s_1 and s_2: the cost of transitioning from frame s_2 to s_1, $C(s_1, s_2)$ is calculated as the Euclidean distance between two (24 coefficient) bark spectrum vectors l and f – where $s_{1,l}$ is the last (46.4 ms) frame of s_1 and $s_{2,f}$ is the first (46.4 ms) frame of s_2:

$$C(s_1, s_2) = \sum_{n=1}^{N=24} \left| s_{1,l}(n) - s_{2,f}(n) \right| \qquad (2)$$

Bark coefficients are used since they are shown to capture both pitch and timbral discontinuities [11].

4. GENERATION

We now detail the generation of arbitrarily long audio sequences in SEED, which rely on the transition table and concatenation cost matrix C detailed in Section 3. Generation is approached as a search problem, which aims at finding audio segment sequences which: i) exist as observable audio class series in the input audio signal, ii) have low concatenation cost, and iii) promote variation and a uniform exploration of the entire set of audio segments.

In order to satisfy the first condition we retrieve from the transition table a list of all possible audio classes that follow the last played audio classes and then unpack them into the audio segments they represent. From this collection of candidate audio segments, we then select the one with minimum concatenation cost C to the preceding segment, thus, favoring smoother transitions in the generated sequences.

Prior to the selection process we introduce a penalty in the concatenation cost of the most recently selected segments in order to prevent repetition (or promote variation). The concatenation costs C of the most recently played segment in the matrix is increased by a factor of $2 \leq \eta \leq 4$. Penalties are gradually reset to the original concatenation cost in eight segment selections by imposing a difference of $(\eta - 1)/8$ at each iteration. Selected segments are concatenated with a short cross-fade overlap of 46.4 ms.

Following the example shown in Figures 3 and 4, and assuming that we want to select an audio segment to follow the audio segment 1, we would first inspect all possible continuations for its audio class by retrieving transitions in the table illustrated in Figure 4, which would result in the single audio class **2**. Then, we would collect all audio segments linked to that audio class, which gives the set {2, 3, 4, 5, 9}. Lastly, we would select for playback the audio segment with the smallest concatenation cost from the previous audio segment 1.

5. USER INTERFACE

Figure 5 shows the graphical user interface of the SEED prototype implemented in Pure Data. On the upper part of the interface we find three elementary control settings for i) opening an input audio signal ('/'), ii) starting and pausing the generative process ('>'), and iii) controlling the overall volume of the generated output (top-right corner slider).

The lower part of the interface contains a function graph that plots the height function of the cluster hierarchies, i.e. the Euclidean distance between the linked pair of elements at each new iteration or the height of each hierarchy in the dendrogram representation. Below the graph, a slider with the same number of elements as the height function allows the user to specify the hierarchy of the clustering algorithm (or the cutting point in the dendrogram), which drives the dictionary construction by defining the number of audio classes to be adopted.

Figure 5. SEED graphical user interface.

The right-most option in the slider results in a dictionary composed of the same audio classes as the number of audio segments S, and will lead to the generation of sequences that correspond to the original input audio signal, thus no novelty is introduced. The left-most option in the slider results in a dictionary with a single audio class that encompasses all audio segments. In the latter case, the generated audio segment sequences result from uniformly-distributed random decisions. Within these extreme cases all clusters ranging from 1 to S can be selected, biasing the generation process towards a higher degree of novelty (i.e., less redundancy).

A height function expressing the similarity h between linked pairs of sound segments or cluster segments at each agglomerative clustering hierarchy q tends to assume an elbow-like shape due to the likelihood of environmental sounds to have a highly redundant content [8]. Thus, we consider the elbow point of this function a balanced solution between the novelty and redundancy in the signal content for driving the dictionary construction, and assign it as the default value once an audio input source is loaded. To compute this point, q^*, we find the point closest to $(1, h_1)$:

$$q^* = \text{argmin}_q \sqrt{(q-1)^2 + {h_q}^2} \qquad (3)$$

Figure 6 shows the height function from the data in Figure 2, from which we compute the elbow point by finding the minimum distance to $(1, h_1)$.

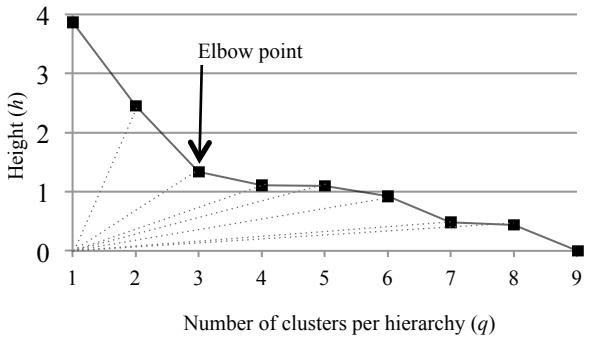


Figure 6. Height function for the different clusters hierarchies shown in Figure 2. Dashed segments indicate the distance to $(1, h_1)$ used to determine the elbow point.

The specification of a cluster hierarchy can be obtained in real-time since the subsequent transition table construction is very fast to compute. This allows users to easily explore different dictionaries of audio classes for a single input audio signal. Additionally, by adopting a manual strategy to fine-tune the audio similarity tolerance degree we avoid the formalization of contextual and subjective criteria inherent to the task.

6. EVALUATION

To evaluate our system SEED, we conducted an experiment to assess whether it can create natural sounding environmental sounds from examples. The experiment consisted of an online listening test, which aimed to i) compare SEED and related state-of-the-art systems in terms of the naturalness of their generated output and ii) assess the naturalness of SEED's output with complex input audio sources. Our hypothesis is that user ratings of SEED's generated examples will be higher than related systems and the difference between original and SEED complex examples are in line with the difference between the original and SEED simple examples, given the enhanced capacity of SEED to model the input signal structure, while enforcing smooth segment transitions.

6.1 Experiment Design

To compare SEED with related state-of-the-art systems in terms of the naturalness of their generated output, we conducted an online listening experiment based on the design and audio dataset used in the evaluation of Fröjd's and Horner's system for the resynthesis of environmental sounds [7]. This dataset consists of eight sonic environments with variable duration ranging from 3 to 23 seconds (see second column of Table 2), which cover a wide range of environmental sounds, whose source descriptions are listed in the first column of Table 2.

Besides the original files, the dataset includes generated output from three systems by: Dubnov et al. [5], Lu et al. [6], and Fröjd and Horner [7]. From the Fröjd and Horner system [7], one resynthesized audio example is provided for each source file. From those by Dubnov et al. [5] and Lu et al. [6] only a few renditions exist. Although a fair comparison to the two latter systems cannot be established, we include them in the listening test following a similar design principle as Fröjd and Horner [7].

To this dataset we added four more audio examples in order to assess the naturalness of SEED's output under more complex audio input signals, which oppose the simple sound textures of the first eight examples. Complexity in these four examples is expressed by the presence of highly dense sonic environments with overlapping events (e.g., sound sources *9. Battle* and *10. Square*), and in the presence of audio inputs with long-term temporal event dependencies (e.g. sound sources *11. Factory* and *12. Gallop*). Input signals with a high density of events can pose additional difficulties in segmentation and concatenation phases, and the long-temporal dependencies in modeling the temporal dimension of the input signal.

All generated audio examples have the same duration of their original audio source, which were also included in the listening test to allow a comparison between the effectiveness of SEED under more and less complex examples. Generated audio in SEED uses a clustering hierarchy computed automatically by the elbow method detailed in Section 5.

Participants were asked to rate on a 7-point Likert scale (1-7) *the naturalness of the sonic environments* (including source and generated audio), where 1 corresponds to highly unnatural and 7 to highly natural. To rate a given audio example participants were asked to listen to its entire duration using high quality headphones. To allow the participants to familiarize themselves with the experiment and, a short training phase was added to the listening test. To prevent response bias introduced by order effects, the musical examples were presented in a random order at each experiment trial. To submit their ratings and complete the listening test, the participants were obliged to rate all sound examples. Participants were not paid to take the experiment.

6.2 Results

In total, 20 subjects (14 female and 6 male) ranging in age from 19 to 41 years old ($mean = 28$ and $standard\ deviation = 7$) participated in the experiment. Four participants claimed to have *high expertise in sound design*, 12 claimed *some expertise in sound design*, and the remaining 4 *no expertise in sound design*. No participants declared hearing problems.

To examine the results of the listening test we show the average ratings per musical example for each system and original source (Table 2). The average and standard deviation (SD) ratings per system are shown in bold. Empty cells in Table 2 correspond to examples not included in the dataset. The average ratings per system largely concur with those of the experiment conducted by Fröjd and Horner [7] and the overall mean rating of SEED is higher than the compared systems. Yet, for the Fröjd and Horner and SEED systems, from which we could compare the entire set of available sound examples, a two-tailed t-test shows no statistically significant difference ($p = .2716$). Furthermore, all systems are rated lower than the original source files in terms of naturalness for most sonic environments (the difference between the original source files and Fröjd and Horner and between the original source files and SEED systems are statistically significant ($p < .0001$).

A deeper examination of the average subjective ratings of the first eight examples is shown in Figure 7. We compare the input source ratings to the systems for which we have the entire set of examples, i.e. Fröjd and Horner [7], and SEED. While the average ratings for the SEED system in the examples 2, 4, 5, and 7 are higher than those by Fröjd and Horner and are comparable with the average ratings of the naturalness of the audio input source. In the remaining four examples (1, 3 6, and 7) the average rating of the Fröjd and Horner are higher.

From the SEED examples with ratings lower than Fröjd and Horner, 1, 6, and 7 have almost identical ratings and audio example 3 has a noticeable difference. Listening back to excerpt 3, we can identify an obvious lack of variation in the example generated by SEED. In this (as in other examples) generated by Fröjd's and Horner's system, the clear overlap of audio chunks contributes to generate a larger degree of variation in simple textural environmental sounds. On the other hand, SEED seems to provide more compelling results for audio input sources that have clearly identifiable patterns (such as in examples 2 and 5).

Figure 8 shows the average naturalness ratings of the four audio input sources included in the listening test to assess the behavior of SEED under audio input signals with higher complexity. SEED ratings are uniformly lower than the input source signals, which is in line with the results from the less complex initial 8 examples. A significant decrease of 18% ($p < .0001$) of naturalness ratings between input sources and SEED output in the set of simples examples, follows a similar tendency in the set of complex examples, which have a significant decrease of 22% ($p < .0001$). Thus, we can conclude that SEED performs in a similar way under audio input sources with different degrees of complexity (as measured in terms of high density and long-term temporal dependencies of audio events).

		Average subjective ratings				
Scene description	Duration	Input Source	Lu et al.	Dubnov	Fröjd Horner	SEED
1. Aviary	9	4,80			2,85	2,80
2. Baby	14	5,75		3,20	3,25	4,40
3. Racing	16	5,15		1,90	4,40	2,90
4. Rain	3	4,40	3,65		2,90	4,25
5. Seagulls	15	5,05			4,15	5,20
6. Shore	19	4,60		1,85	4,70	4,25
7. Stream	5	3,70	3,65		3,05	3,45
8. Traffic	23	4,40		2,60	3,80	3,65
Average	**13**	**4,73**	**3,65**	**2,39**	**3,64**	**3,86**
SD	**6,87**	**1,729**	**2,05**	**1,84**	**1,83**	**1,82**
9. Battle	38	3,70				3,10
10. Square	45	6,10				4,30
11. Factory	59	5,40				4,60
12. Gallop	20	4,80				3,60
Average	**40**	**5,00**				**3,90**
SD	**16,22**	**1,47**				**1,67**

Table 2. Participant's ratings of the naturalness of 12 sonic environments extended in its original and resynthesized versions by four different systems.

7. CONCLUSIONS AND FUTURE WORK

In this paper, we presented SEED, an application for generating variably long audio renderings of recorded environmental sounds while preserving their inherent structure. Our method introduces two main novelties in relation to state-of-the-art systems [3-7]. The first is to use concepts from concatenative sound synthesis to decouple structurally segmented audio into two moments (middle region and boundaries). Features extracted from segment boundaries are used to measure the concatenation cost between audio segments. Features extracted from the middle region of the segment are used to build a temporal model of audio signal input.

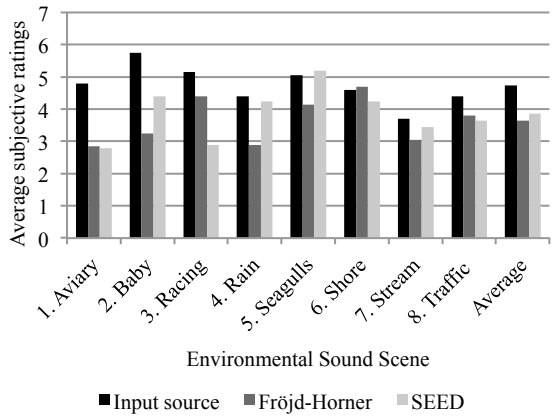


Figure 7. Average subjective ratings of the naturalness of eight (simple) environmental sound scenes, each including an original input audio source and two resynthesized versions by Fröjd and Horner and SEED.

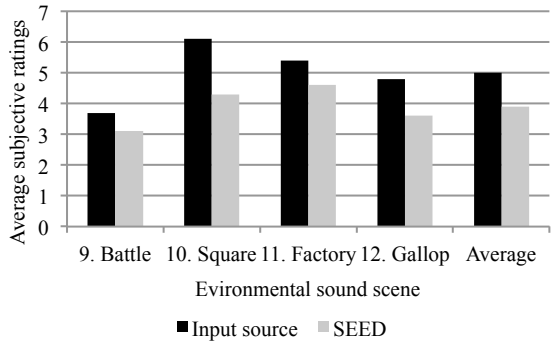


Figure 8. Average subjective ratings of the naturalness of four (complex) environmental sound scenes, each including an original input audio source and a resynthesized version by SEED.

The second contribution of our work is the use of a hierarchical clustering algorithm to reduce the dimensionality of segmented audio data into a discrete and finite dictionary of audio classes. This representation is then used to create a transition table, which encodes the audio input signal temporal structure. Based on the transition table we can then generate patterns that are similar to those in the input audio signal with variable degrees of variation.

Many potential applications exist for our work. Among these we can highlight: the modification and extension of recorded sounds to fit given scenes in sound design; the generation of ever-changing environmental sounds in games using limited memory storage; the restoration of audio signals (e.g. in the loss of audio or musical packets transferred over the Internet), and even in audio compression.

Through our preliminary evaluation, we have shown that SEED generates natural sounding environmental soundscapes to a degree that outperforms related state-of-the-art systems and that audio input signals with dense and complex temporal structures were similarly rated in relation to the original input audio signals. Furthermore, we hope SEED's interface encourages experimentation towards optimal results. Several examples generated by SEED, including those from the experiment are available online at: `http://bit.ly/29vnEhc`.

In future work, we will strive to assess the degree to which the parameter setting biases the generation as well as the best set of parameters in accordance to the type of input audio signals. In greater detail, we will further study the implication of the following parameters: duration of the input file, number of audio segments, entropy and redundancy in the audio input signal and its implications in the definition the number of classes in the dictionary, and the order of the transition table in relation to the all aforementioned settings. Furthermore, we plan to design and conduct an evaluation experiment in which generated audio examples exceed the duration of the input audio source. In this way we can then explicitly assess the extent to which a given file can be extended, and thus validate the ultimate goal of the application.

We will also study the use of audio descriptors that express information concerning the spatial-temporal content of the audio as a strategy to drive the resynthesis process aiming to further enhance temporal coherence of the meso structure of the generated output.

Acknowledgments

Project "TEC4Growth - Pervasive Intelligence, Enhancers and Proofs of Concept with Industrial Impact/NORTE-01-0145-FEDER-000020" is financed by the North Portugal Regional Operational Programme (NORTE 2020), under the PORTUGAL 2020 Partnership Agreement, and through the European Regional Development Fund (ERDF). This research is also supported by the Portuguese Foundation for Science and Technology under the post-doctoral grant SFRH/BPD/109457/2015.

8. REFERENCES

[1] D. Sonnenschein, Sound Design. Michael Wiese Productions, 2001.

[2] R. Viers. Sound Effects Bible. Michael Wiese Productions, 2011.

[3] D. Keller and B. Truax, Ecologically based granular synthesis. Proceedings of the International Computer Music Conference, 1998, pp. 117-120.

[4] R. Hoskinson and D. Pai, "Manipulation and Resynthesis with Natural Grains," in Proceedings of the International Computer Music Conference, 2001, pp. 338-341.

[5] S. Dubnov, Z. Bar-Joseph, R. El-Yaniv, D. Lischinski, and M. Werman, "Synthesizing Sound through Wavelet Tree Learning," in IEEE Computer Graphics and Applications, 22(4), 2002, pp. 38-48.

[6] L. Lu, L. Wenyin, and H.J. Zhang, "Audio Textures: Theory and Applications," in IEEE Trans. on Speech and Audio Processing, 12(2), 2004, pp. 156-167.

[7] M. Fröjd and A. Horner, "Fast Sound Texture Synthesis Using Overlap-add," in Proceedings of the

International Computer Music Conference, 2007, pp. 320-323.

[8] G. Strobl, G. Eckel, D. Rocchesso, and S. le Grazie, "Sound Texture Modeling: A Survey," in Proceedings of the Sound and Music Computing Conference, 2006, pp. 61-65.

[9] D. Schwarz, "State of the Art in Sound Texture Synthesis," in Proceedings of the Digital Audio Effects Conference, 2011, pp. 221-231.

[10] G. Bernardes, C. Guedes, and B. Pennycook, "Eargram: An Application for Interactive Exploration of Concatenative Sound Synthesis in Pure Data," in LNCS, 7900, 2013, pp. 110-129.

[11] W. Brent, "A Timbre Analysis and Classification Toolkit for Pure Data," in Proceedings of the International Computer Music Conference, 2009, pp. 224-229.

[12] J. Buys, Generative Models of Music for Style Imitation and Composer Recognition. Honours Project in Computer Science, University of Stellenbosch, 2011.

[13] D. Mitrovic, M. Zeppelzauer, and H. Eidenberger, "Analysis of the Data Quality of Audio Descriptions of Environmental Sounds," in Journal of Digital Information Management, 5(2), 2007, pp. 48-54.

[14] D. Keller and J. Berger, "Everyday sounds: Synthesis parameters and perceptual correlates," in Proceedings of the VIII Brazilian Symposium on Computer Music. Fortaleza, CE: SBC, 2001.

[15] G. Bernardes, Composing Music by Selection: Content-Based Algorithmic-Assisted Composition. Ph.D. thesis, University of Porto, 2014.

[16] B. S. Everitt, Cluster Analysis. Edward Arnold, 1993.

[17] N. Saint-Arnaud and K. Popat, "Analysis and Synthesis of Sound Texture," in D. F. Rosenthal, Horoshi G. Okuno (editors), Computational Auditory Scene Analysis. New Jersey, NJ: Lawrence Erlbaum Association, 1998.

SONIFICATION OF DARK MATTER: Challenges and Opportunities

Núria Bonet
Interdisciplinary Centre for Computer Music Research, Plymouth University, Plymouth PL4 8LY (UK)
nuria.bonet@plymouth.ac.uk

Alexis Kirke
alexis.kirke@plymouth.ac.uk

Eduardo R. Miranda
eduardo.miranda@plymouth.ac.uk

ABSTRACT

A method for the sonification of dark matter simulations is presented. The usefulness of creating sonifications to accompany and complement the silent visualisations of the simulation data is discussed. Due to the size and complexity of the data used, a novel method for analyzing and sonifying the data sets is presented. A case is made for the importance of aesthetical considerations, for example musical language used. As a result, the sonifications are also musifications; they have an artistic value beyond their information transmitting value. The work has produced a number of interesting conclusions which are discussed in an effort to propose an improved solution to complex sonifications. It has been found that the use primary and secondary data parameters and sound mappings is useful in the compositional process. Finally, the possibilities for public engagement in science and music through audiences' exposure to sonification is discussed.

1. INTRODUCTION

The Sonification of Dark Matter is an audiovisual work composed from the sonification and visualization of dark matter simulation data. Dark matter does not absorb or emit radiation, it is therefore invisible. We can only perceive its effects on the baryonic (visible) matter. In fact, we know that about 95% of the universe should be made up of dark matter and dark energy in order to justify the behavior of the universe according to the laws of physics. The gravitational forces in the universe are far too large if we accept that only baryonic matter is acting on them. The standard model of structure formation contains enough dark matter and dark energy to explain these gravitational effects. The simulations of dark matter such as those described here can be compared with observational data of the universe such as galaxy surveys [1]. Cognitively speaking, dark matter is not perceivable to us does not emit sound signals either. Therefore, dark matter poses a particular challenge when attempting to visualize and sonify it. The particle data used were created and visualized by Kaehler, Hahn and Abel thanks to a novel visualization method for N-body simulations [1]. A Max/MSP based implementation of the sonification of the data provides additional cognitive bandwidth for understanding the behavior of dark matter. It also increases the audience's attention engagement with the originally silent visualisations, as we have come to expect 'sound to accompany animated images' [2]. By increasing the sensory dimensions of the display, the effectiveness of information transmission is increased.

The aesthetical requirements for a successful sonification are discussed, as well the compositional opportunities resulting from it. Furthermore, the challenges of interdisciplinary music composition with sonification are identified. *The Sonification of Dark Matter* is presented as a practical example of the implementation of a sonification algorithm for dark matter.

2. RELATED WORKS

The sonification of particle physics data has been of increasing interest to scientists, particularly since the discovery of the Higgs Boson particle by CERN [3] and the gravitational waves by LIGO [4]. They have proven to be popular with the general public and scientists, for their accessibility and novelty factor respectively. The Quantizer project provides a platform for real-time sonification of the ATLAS experiment at CERN, where the listener can choose a musical style in which the sonification is mapped [5]. While these examples prove method's potential for public engagement, they do not exploit the full musical and therefore cognitive potential of sonification. In fact, they are heavily based on ideas of pitch, rhythm and harmony, as well as Western Classical musical styles. We suggest that the inclusion of further musical parameters such as timbre, spatialisation and volume can help harness the full potential of sonification. Vogt and Höldrich explore the idea of metaphoric sonification where intuitive mappings for better interpretation of sound and particle data [6]; their method has been applied to the ALICE experiment at CERN [7].

3. SONIFICATION

The intended outcome for the sonification project was an audio-visual work, the data sets was always intrinsically related to its visual counterpart. Considering the sonification as a 'concert piece' means that *musification* would be a more accurate term of the method, as it describes sonifications used 'for artistic purposes' [8]. Beyond the conscious decision of presenting and listening to a sonification as music, a musification should also use elements of the sonification process to define elements of the music. All too often, the musical thought is reduced to choice of sound mappings and instrumentation. By structuring the piece according to the structure of the data for example, the piece is not only a translation of the data but *is* the data. Therefore, if the data is organized as to create a musical structure, we are making compositional choices towards a musification. While we believe that the resulting audio-visual product could be enjoyed purely for aesthetic reasons without the knowledge of the underlying data and processes, the experience of the audience is enriched by the knowledge of its conception.
The elements that contribute to a successful sonification can be summarized by four categories: the choice of data, the choice of sound mappings, the choice of musical language and the emotional content of the data and the sonification. The combination of these parameters is unique to each data set thus there is no unique solution to data sonification [9]; only a careful combination can really transmit the information and emotional content appropriately

3.1 Data

The data used for sonification were the same as used for visualisation by Kaehler et al. [1]. The data sets are very large, e.g the simulation called 'Warm Dark Matter Halo' tracks 17 million particles which results in over 100 million tetrahedra per time step; another simulation discussed by Kaehler et al. contains 'about 134 million particles, resulting in about 804 million tetrahedra, respectively 3.2 billion triangles' [1]. Due to the size and complexity of the data sets, it was therefore imperative to filter and analyse them in order to highlight interesting patterns showing to physical phenomena. Three data sets and their respective visualisations were used for sonification; 'Warm Dark Matter Halo', 'Dark Matter Streams' and 'Dark Universe' [10]. Respectively, they simulate the formation of a dark matter halo around a galaxy, dark matter forming streams and the spatial concentration of dark matter in the universe.

The size and complexity of the data sets means that a second-order sonification is needed to express complex relationships in the data. Gresham-Lancaster describes the second-order sonification as the 'application of time bound algorithmic processes that are driven by sets or clusters of a data set' [11]. The 'sets or clusters' representing interesting relationships and patterns in the data were extracted by analyzing the simulation visualization.

3.2 Mappings

Primary and secondary data parameters were translated to primary and secondary sound mappings. 'Primary cues' were attributed to sound parameters to which we are particularly sensitive and capable of perceiving even small changes [12]; in this case pitch frequency and rhythm. Ballora describes 'supporting auditory cues' which underline the distinction between different primary cues while being more difficult to perceive; interesting examples are panning and volume. While this differentiation is useful for sonification design, it also has an interesting relation to musical composition where parameters have different cognitive levels but contribute to the overall musical structure. The hierarchy in parameters is thus a crucial element in the composition of a musification; to create a sonification which represents the data but is also structurally determined by the data.
In practice, examples of data parameters we identified were the size of a dark matter halo or galaxy, spatial particle concentration and distribution, and the movement of structures through space. As previously discussed, the data parameters used were of a higher-level nature where the relationships between the data points are explored rather than the data itself. These were mapped to sound parameters such pitch and rhythm, but also volume, spatial panning and timbre; the use of data parameters to trigger sound events was also explored.

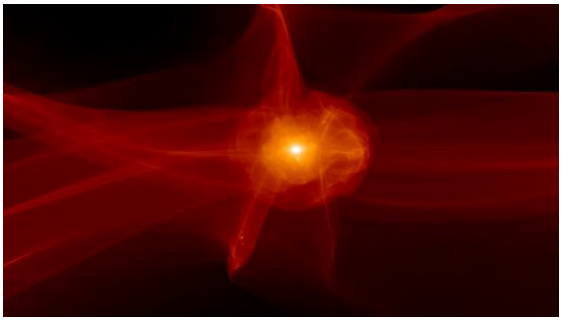


Figure 1. Visualization of a warm dark matter halo simulation. Visualization by Ralf Kaehler and Tom Abel, simulation by Oliver Hahn and Tom Abel [10].

3.3 Musical Language and Emotional Content

We know that sonification can be a powerful tool when used to harness the 'perceptual strengths that we possess as human beings' [9]. It is then surprising and counter-productive that sonification design so often relies on a well-established but limited set of sound parameters within the Western Classical musical language: pitch, harmony and rhythm. While these are understood almost universally and can be very effective, further musical parameters should also be considered for full effect. The electroacoustic or sound-based musical language focuses on sound qualities such as timbre, spatial positioning, frequency spectrum and so on; we feel that a combination of musical languages can broaden the cognitive possibilities of sonification. The discrete nature of the musical scale is hardly appropriate to describe continuous data. For example, the mapping

of tiny differences of data to an audible frequency can be very effective as humans can distinguish tiny changes in frequency. We can see, that the variety of types of data demands that we broaden the 'toolkit' and that we adapt the mappings to the data. Note that these suggestions are in line with the development of music composition over the last century, during which composers have increasingly turned towards sound-based music and transcended the restrictive and discrete nature of Classical Western musical language.

Creating an emotional connection between the listener and the sonification can be another way of engaging them in the data. Vogt and Höldrich speak of 'metaphorical sonification' [6], the author prefers the concept of 'empathetic sonification' [12]. That is a sonification which 'engages the listener's ears and emotions in equal measure'. This is achieved by a sonification which musically reflects the emotional meaning of the data intended by the composer; the potential soundscape of the work should also reflect the data. However, the cognitive void presented by dark matter poses a challenge to this approach to sonification. A quick conclusion would be to assimilate sounds of space crafts or sounds associated to science-fiction with the sound of dark matter. This is problematic for a number of reasons. While dark matter exists in the universe and therefore has a link to space exploration, it also exists on earth. Furthermore, the music and sounds of science-fiction are ultimately a social construct of what space exploration sounds like.

While circumventing the danger of relying on stereotypical soundscapes and sonic associations in order to evoke the emotional content of the data, the listener's innate and learnt connotations can also be harnessed to transmit a more powerful message. *The Sonification of Dark Matter* uses a mixture of Western Classical and electroacoustic musical languages for this reason. The sound sources are recordings of a piano and an array of synthesizers built in Max/MSP. Finally, some elements of we what might consider a space or science-fiction soundscape were included in the final sonification. That is to say, the use of synthesized sounds evoked these connotations for a large number of audience members. The listeners found this reference rather fitting as it provided a musical reference in what is otherwise a complex subject matter and resulting musification.

3.4 Methods

The visualisations were analyzed in Max/MSP to identify data clusters; this is because the size and complexity of the data made this approach far more effective than to deal with the raw data. By adjusting parameters of brightness, contrast and saturation of the image, specific patterns emerged which could be regarded as the visual filtering of the data. Subsequent RGB analysis calculated the amount of a specific color or its position in the image and therefore revealing the concentration and spatial mapping of particles in the image. As the colors in the visualisations correspond to the concentration of particles – the visualization effectively made artistic choices in choosing the colors - the RGB analysis resulted in a filtered and parsed data set. The data clusters were then mapped to sound parameters as previously described. The implementation of the RGB analysis and the second-order sonification in Max/MSP allowed us to produce sonifications in real-time, this was particularly useful for monitoring the results.

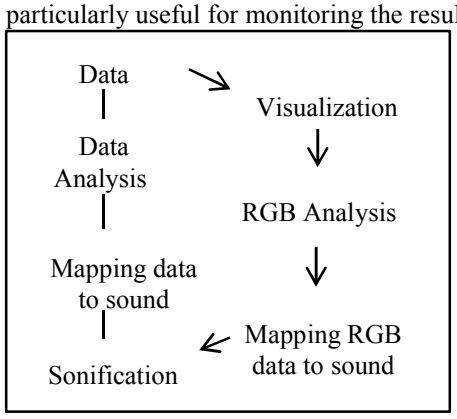


Figure 2. Process from data to sonification. The left side shows the typical method for data sonification, the right side shows the modified method needed to deal with dark matter simulation data.

3.5 Presentation

The project the piece *The Sonification of Dark Matter*, a 17-minute movie which accompanies the listener through the process of the sonification of the data from the silent visualisations to the completed sonifications in conjunction with their corresponding visualisations. As such, the final work serves as a tool to introduce the beginner to dark matter but also the process of sonification. It was felt that this was crucial for listeners to understand the full complexity of the original data, the method of sonification and the resulting audiovisual work. *The Sonification of Dark Matter* was premiered at the *Peninsula Arts Contemporary Music Festival* at Plymouth University (Plymouth, UK) on 26-28 February 2016.

4. DISCUSSION

The work with large-scale data sets of dark matter simulations highlights the challenge of using sonification as a compositional tool. The method demands musical and scientific knowledge in order to be successful; it can at times be impossible for a composer to fully understand the scientific background of the data sonified without the collaboration of a subject specialist. The discrepancy between the two knowledges has been mostly highlighted in sonification for scientific use which is often aesthetically poor and possibly unpleasant. A lack of aesthetic understanding of auditory display can render a sonification unappealing to listeners but also relay little information to the listener. On the other hand, a musification which transmits information poorly can be appreciated as a musical work but loses its purpose as a sonification. Therefore, a scientific and musical collaboration is indispensable to allow

the field to progress and tackle more complex data and music.

Some sound mappings were more effective than others. As previously mentioned, pitch and rhythm parameters were particularly effective as listeners could perceive even slight variations; the use of volume was far less effective in that sense. However, the combination of the primary parameters (pitch and rhythm) with secondary parameters (volume) was crucial in creating a satisfying musification and transmitting information. In fact, the hierarchy of parameters established for the sonification mirrored the used of primary and secondary parameters in music composition. The structure of the data also informed the creative process as it determined the structure of the music. Finally, a careful choice of musical parameters and languages created a powerful piece. These elements elevated the aural display to musification.

The positive response from audiences showed that sonification can be used in public engagement in both music and science. By presenting a number of different mappings to the audience before collating them into a final version, the listeners could learn about sonification while also understanding the compositional process behind the final product. Effectively, they were involved with and gained an understanding of contemporary and electroacoustic music through the audiovisual installation. They were also introduced to the concept of dark matter in an accessible manner and had the chance to experience different aspects of the phenomenon in an educational environment. We found that the installation engaged crowds which were interested in the science aspect of the project in music, and vice-versa. In conclusion, the use of sonification and musification has widened the participation in scientific and musical outreach.

Acknowledgements

This work was made possible by the support of Ralf Kaehler at the Kavli Institute for Cosmology and Astrophysics, Stanford University, US.

5. REFERENCES

[1] R. Kaehler, O. Hahn, and T. Abel, “A Novel Approach to Visualizing Dark Matter Simulations”, in IEEE Transactions on Visualization and Computer Graphics, 2012, pp. 2078-2087.

[2] C. Scaletti, and A.B. Craig, “Using sound to extract meaning from complex data”, in Proc. SPIE, 1991.

[3] https://lhcopensymphony.wordpress.com

[4] http://www.ligo.org/multimedia.php

[5] http://quantizer.media.mit.edu/

[6] K. Vogt, and R. Höldrich, “A metaphoric sonification method – towards the acoustic standard model of particle physics”, in International Conference on Auditory Display, 2010, pp. 271-278.

[7] K. Vogt, R. Höldrich, D. Pirrò, M. Runori, S. Rossegger, W. Riegler, M. Tadel, “A Sonic Time Projection Chamber: Sonified Particle Detection at CERN”, in International Conference on Auditory Display, 2010.

[8] F. Grond and J. Berger, “Parameter Mapping Sonification”, in T. Hermann, A. Hunt, and J.G. Neuhoff, The Sonification Handbook. Logos Publishing House, 2011, ch. 15, pp. 363-397.

[9] M. Ballora, “Sonification, Science and Popular Music: In search of the ‘wow’”, in Organised Sound, 2014, pp. 30-40.

[10]https://www.slac.stanford.edu/~kaehler/

homepage/visualizations/visualizations.html

[11] S. Gresham-Lancaster, “Relationships of sonification to music and sound art”, in AI&Soc, 2012, pp. 207-212.

[12] N. Bonet, A. Kirke, and E.R. Miranda, “Blyth-Eastbourne-Wembury: Sonification as a tool in electroacoustic composition” in Proc. Int. Conf. New Music Concepts (ICNMC 2016), Treviso, 2016.

MELODY EXTRACTION BASED ON A SOURCE-FILTER MODEL USING PITCH CONTOUR SELECTION

Juan J. Bosch
Music Technology Group,
Universitat Pompeu Fabra, Spain
juan.bosch@upf.edu

Emilia Gómez
Music Technology Group,
Universitat Pompeu Fabra, Spain
emilia.gomez@upf.edu

ABSTRACT

This work proposes a melody extraction method which combines a pitch salience function based on source-filter modelling with melody tracking based on pitch contour selection. We model the spectrogram of a musical audio signal as the sum of the leading voice and accompaniment. The leading voice is modelled with a Smoothed Instantaneous Mixture Model (SIMM), and the accompaniment is modelled with a Non-negative Matrix Factorization (NMF). The main benefit of this representation is that it incorporates timbre information, and that the leading voice is enhanced, even without an explicit separation from the rest of the signal. Two different salience functions based on SIMM are proposed, in order to adapt the output of such model to the pitch contour based tracking. Candidate melody pitch contours are then created by grouping pitch sequences, using auditory streaming cues. Finally, melody pitch contours are selected using a set of heuristic rules based on contour characteristics and smoothness constraints. The evaluation on a large set of challenging polyphonic music material, shows that the proposed salience functions help increasing the salience of melody pitches in comparison to similar methods. The complete melody extraction methods also achieve a higher overall accuracy than state-of-the-art approaches when evaluated on both vocal and instrumental music.

1. INTRODUCTION

The task of melody extraction from polyphonic music recordings has been generally approached with salience-based or separation-based methods [1]. Salience-based approaches compute a frame-based pitch salience function, while separation-based approaches attempt to isolate the melody source from the mixture more or less explicitly. Melody oriented pitch salience functions should ideally only contain a peak at the frequency corresponding to the melody pitch present at a given instant.

The most commonly used pitch salience function is harmonic summation [2]. This approach is computationally inexpensive and has been used successfully in a variety of forms for predominant melody extraction [3, 4] or multiple pitch estimation [5]. More recently, probabilistic approaches based on decomposition models such as Non-negative Matrix Factorisation (NMF) have gained more interest [6, 7], especially within source separation scenarios.

Salamon and Gómez [3] propose a salience function based on harmonic summation [2], computed as the sum of the weighted energies found at integer multiples (harmonics) of each of the considered frequencies. Durrieu et al. [6] propose a salience function within a separation-based approach using a Smoothed Instantaneous Mixture Model (SIMM), as detailed in Section 2. There are important differences between the salience functions obtained with SIMM (H_{f_0}) and harmonic summation (HS). H_{f_0} is much more sparse, and has a larger range of values, since the method does not prevent values to be very high or very low. Figure 3 shows a comparison of both salience functions for one of the excerpts used for evaluation: (a) shows the pitch salience function obtained with SIMM, and (b) corresponds to HS, which is more dense and smooth, and has a smaller range of values.

Melody extraction methods exploit salience functions for pitch tracking, relying on the energetic predominance of melody pitches and on melody contour smoothness, using e.g. streaming rules [4] Hidden Markov Models (HMM) [7, 8], or pitch contour characteristics [3]. Finally, frames are classified as voiced or unvoiced (containing a melody pitch or not respectively), using static or dynamic thresholds [4, 7], or exploiting pitch contour salience distribution [3]. For instance, Durrieu et al. [8] use an empirically chosen fixed threshold, such that voiced frames represent more than 99.95% of the leading instrument energy. Salamon [9] proposed a generative model to distinguish melody from non-melody contours, and Bittner [10] proposed a discriminative classifier based on contour features. While both approaches learnt from training data, none of them increased the overall accuracy obtained with the method based on heuristic rules [3].

Main challenges in melody extraction deal with more complex music material [1], with melodies played by different instruments, harmonised melodic lines, or music that features "ensemble" sounds, typically found when several performers play or sing in unison. Some characteristics of such sounds is the fluctuation of the pitches, known as voice flutter, typically found in orchestral and choral music [11]. A key step towards the development of more advanced algorithms and a more realistic evaluation are large and open annotated databases. Recent works presented datasets for melody extraction with such char-

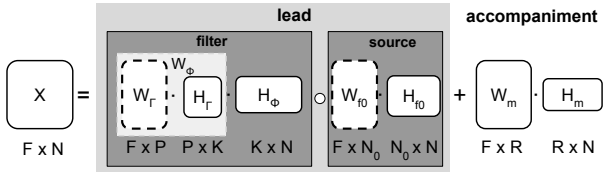


Figure 1: SIMM model. Dashed lines refer to the matrices which are fixed, while the rest are iteratively estimated

acteristics, e.g. in a variety of genres and instrumentation (MedleyDB) [12], and in orchestral music (Orchset) [13]. These datasets allow broader definitions of melody than the one used in Music Information Retrieval Evaluation eXchange (MIREX) [14], since they are not restricted to a single instrument. Results on both datasets generally drop significantly in comparison to results on simpler datasets used in MIREX [12, 13, 15].

Previous works [13, 15] have shown the benefits of using separation-based approaches such as the [8] for pitch estimation on orchestral data. However, voicing detection was identified as a key aspect to improve in their method [8]. In this work, we address some of the mentioned challenges, by combining separation and salience-based methods [16, 17], as presented in Section 2. In Section 3 we present the methodology to evaluate pitch salience functions and melody extraction methods, and we present and discuss the results in Section 4.

2. METHOD

We propose a melody extraction method, based on the combination of a salience function based on a source-filter model [6] with melody tracking based on pitch contour selection (PCS) [3]. Our intention is to obtain both an accurate pitch estimation and a good voicing detection, based on their results in [15], and in MIREX.

We propose two different salience functions which aim at adapting the characteristics of H_{f_0} to a melody tracking stage based on pitch contour selection. The first salience function (CB) combines two salience functions: one based on SIMM (H_{f_0}) [6, 8] and another one based on harmonic summation (HS) [3]. The second approach (EW) uses an estimate of the energy of the melody. Both approaches employ Gaussian filtering, since we hypothesise that such smoothing is useful to make melody pitches more salient, particularly in the case of "ensemble" sounds.

We reuse code from Durrieu's source-filter model implementation [1] and Essentia [2] [18], an open source library for audio analysis with a slightly different implementation of [3] compared to MELODIA [3] . Our source code is available for research reproducibility [4] .

2.1 Pitch salience function based on SIMM

Following [6], we model the spectrum of the signal as the lead instrument plus accompaniment: $\hat{X} = \hat{X}_v + \hat{X}_m$,

[1] https://github.com/wslihgt/separateLeadStereo
[2] http://essentia.upf.edu
[3] http://mtg.upf.edu/technologies/melodia
[4] https://github.com/juanjobosch/SourceFilterContoursMelody

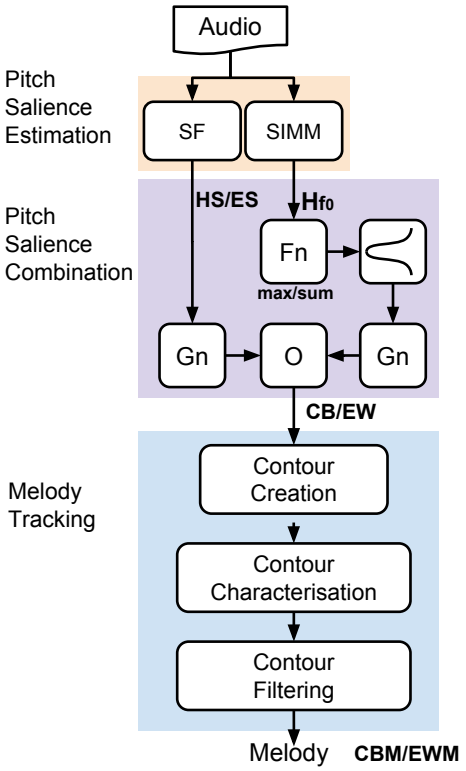


Figure 2: **Left**: Proposed method schema. SIMM: Smoothed Instantaneous Mixture Model (outputs H_{f_0}); SF: salience function, either Harmonic Summation (outputs HS) or Energy-based Salience (outputs ES); Fn: Frame-wise normalisation; Gn: Global normalisation; o: Hadamard product; Gaussian symbol: Gaussian filtering. Combining H_{f_0} with HS we obtain CB. Combining it with ES we obtain EW. CBM and EWM denote the complete melody extraction methods.

where $\hat{X}$ represents the modelled spectrum. The lead instrument is modelled as: $\hat{X}_v = X_\Phi \circ X_{f_0}$, where X_{f_0} corresponds to the source, X_Φ to the filter, and the symbol $\circ$ denotes the Hadamard product. Both source and filter are decomposed into basis and gains matrices as $X_{f_0} = W_{f_0} H_{f_0}$ and $X_\Phi = W_\Phi H_\Phi$ respectively. The filter basis matrix W_Φ is further decomposed into a weighted sum of smooth spectral atoms: $W_\Gamma H_\Gamma$. H_{f_0} corresponds to the pitch activations of the source, which can also be understood as a representation of pitch salience [6]. The accompaniment spectrum is modelled as: $\hat{X}_m = \hat{W}_m \hat{H}_m$, leading to Equation 1.

$$X \approx \hat{X} = (W_\Gamma H_\Gamma H_\Phi) \circ (W_{f_0} H_{f_0}) + W_m H_m \quad (1)$$

Several parameters of the algorithm need to be specified: the number of bins per semitone (U_{st}), the number of possible elements of the accompaniment (R), the number of atomic filters in W_Γ (K), and the maximum number of iterations (N_{iter}). Parameter estimation is based on Maximum-Likelihood, with a multiplicative gradient method [8], updating parameters in the following order for each iteration: H_{f_0}, H_Φ, H_m, W_Φ and W_m. Figure 1 represents the blocks of the Smoothed Instantaneous Mixture Model.

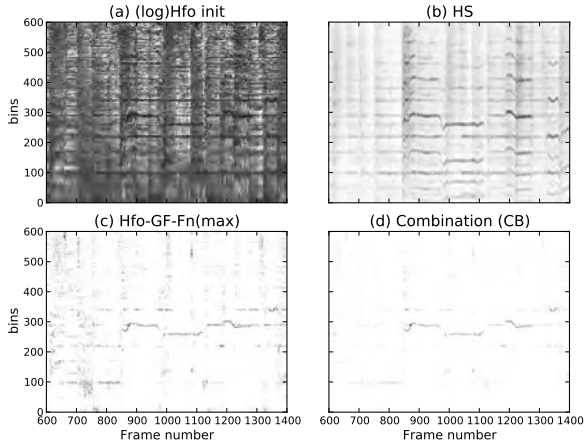


Figure 3: Time-frequency pitch salience representation of an excerpt from "MusicDelta_Beatles.wav" (MedleyDB) with (a) SIMM: $log_{10}(H_{f_0})$ is represented, to reduce the range of values for visualisation purposes) (b) Harmonic Summation: HS (c) H_{f_0} (max) normalised per frame and gaussian filtered (d) Combination (CB).

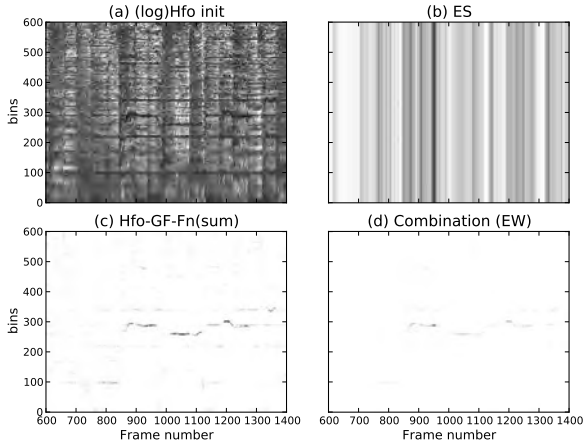


Figure 4: Time-frequency pitch salience representation of an excerpt from "MusicDelta_Beatles.wav" (MedleyDB) with (a) SIMM: $log_{10}(H_{f_0})$ is represented, to reduce the range of values for visualisation purposes) (b) Energy-based matrix: ES (c) H_{f_0} normalised per frame and gaussian filtered (d) Combination (EW).

2.2 CB: Combination with Harmonic Summation

In order to adapt H_{f_0} for pitch contour based tracking, we first propose to combine it with a Harmonic Summation salience function (HS), since pitch contour tracking, was originally adapted to this kind of representation [3]. The computation of HS starts with a Short Time Fourier Transform (STFT) as time-frequency transformation, applies Equal-Loudness Filters (ELF), finds spectral peaks positions and magnitudes, and then refines them using parabolic curve fitting (as implemented in Essentia).

We normalize and combine the considered pitch salience functions $HS(k,i)$ and $H_{f_0}(k,i)$, where *k* indicates the frequency index (bin) and *i* the frame index. The process is illustrated in Figure 2: 1) **Global normalization** (Gn) of HS, dividing all elements by their maximum value $max_{k,i}(HS(k,i))$. 2) **Frame-wise normalization** (Fn) of H_{f_0}. For each frame *i*, divide $H_{f_0}(k,i)$ by $max_k(H_{f_0}(k,i))$. 3) **Convolution in the frequency axis** *k* of H_{f_0} with a Gaussian filter to smooth estimated activations. The filter has a standard deviation of 0.2 semitones. 4) **Global normalization** (Gn), whose output is $\widetilde{H_{f_0}}$ (see Figure 2 (c)). 5) **Combination** by means of element-wise product: $S_c = \widetilde{H_{f_0}} \circ HS$ (see Figure 3 (d)).

2.3 EW: Energy-based normalisation

In order to reduce the range of salience values of H_{f_0}, one possibility would be to simply normalise each frame with the maximum salience. The drawback of this solution is that high salience values also appear in unvoiced frames, which would make voicing detection based on pitch contour selection a complicated task. In order to reduce the salience of unvoiced parts, we employ a frame-wise energy estimate of the melody line, using the method in [8]. For energy estimation, a HMM is employed, where each state corresponds to one bin of the pitch salience function (H_{f_0}), and the probability of each state corresponds to the estimated salience. Pitch continuity is considered in the transition probabilities, favouring smoothness in pitch trajectories. The energy of the melody source for each frame i (E_i), is then computed using the decoded pitch sequence and the matrix decomposition computed before.

The estimated energy is then used to create a matrix (ES) with the same size as H_{f_0}, in which all bins in one frame are equal to the estimated energy in that frame: $ES(k,i) = E_i, \forall k$. ES is then combined with H_{f_0} to create the salience function EW, following the same steps introduced in Section 2.2 (see Figure 2), with the difference that in the frame-wise normalisation (Fn), $H_{f_0}(k,i)$ is divided by $\sum_k H_{f_0}(k,i)$, instead of the maximum value, also following Durrieu's approach. Figure 4 illustrates the combination.

2.4 Melody tracking

From the proposed lead-enhanced salience functions, we create pitch contours by grouping continuous sequences of salience peaks, following [3]. Several parameters need to be set (default values used here are presented between brackets). Salience peaks are first filtered per frame: peaks below a threshold factor τ_+ (0.9) of the highest salience peak are filtered out. Secondly, peaks are filtered if their salience is below $\mu_s - \tau_\sigma \cdot \sigma_s$, where μ_s and σ_s are the mean and standard deviation of the salience of remaining peaks (in all frames). τ_σ (0.9) determines the accepted degree of deviation below mean salience. Contours are created by grouping peaks which are close in time and frequency, with several parameters: the minimum allowed contour duration (100 ms); maximum allowed pitch change during 1 ms time period (27.56 cents) and maximum allowed gap duration (tc = 50 ms). Default parameters here are the same as in [3], except for tc. We analyse the effect of some of these parameters in Section 4.3.

Created contours are then characterised by a set of features: pitch (mean and deviation), salience (mean, standard deviation), total salience, length and presence of vi-

Method	Salience	Description
DUR	H_{f_0}	Source-filter model (SIMM) [8]
SAL	HS	VAMP Implementation of [3]
ESS	HS	Essentia implementation of [3]
CBM	CB	HS+H_{f_0} with PCS
EWM	EW	Energy weighted H_{f_0} with PCS

Table 1: Overview of the evaluated melody extraction methods. PCS: Pitch Contour Selection

brato. Contour features are then exploited for voicing detection, octave error minimisation, and final melody selection. Non-melody contours are filtered out using a voicing detection threshold τ_ν, based on contour salience distribution: $\tau_\nu = \overline{C_s} - \nu \cdot \sigma_{C_s}$ where $\overline{C_s}$ and σ_{C_s} are the contours' salience mean and standard deviation. We focus on the effect of parameter ν (0.2), which controls the amount of filtered contours. For a more detailed explanation, the reader is referred to [3].

Complete melody extraction methods using the proposed salience functions are here denoted as CBM (using CB) and EWM (using EW) (see Figure 2).

3. EVALUATION

We conduct two different kind of evaluation experiments in order to analyse the benefits of combining the proposed salience functions with pitch contour-based tracking. First, the proposed salience functions are evaluated and compared to H_{f_0} and HS in terms of their usefulness for melody extraction. Second, the complete melody extraction approaches are compared to Durrieu et al. [8] (DUR) and two implementations of Salamon and Gómez [3]: VAMP plugin MELODIA (SAL) and Essentia (ESS). Table 1 presents an overview of the evaluated methods. The motivation to conduct the evaluation at two different levels is to better understand the benefits of the combination of salience functions, and the effect on the complete melody extraction method. The pitch resolution (number of bins per semitone) was set to $U_{st} = 10$, and the hop size was 256 samples, except for SAL which is fixed to 128. Sampling rate was 44100 Hz. The frequency limits were set to $f_{min} = 55$ Hz and $f_{max} = 1760$ Hz for all algorithms.

3.1 Datasets

The evaluation is conducted on MedleyDB and Orchset datasets, converted to mono as (left+right)/2. MedleyDB contains 108 melody annotated files (most between 3 and 5 minutes long), with a variety of instrumentation and genres. We use two definitions of melody, **MEL1**: the f_0 curve of the predominant melodic line drawn from a single source (MIREX definition), and **MEL2**: the f_0 curve of the predominant melodic line drawn from multiple sources.

Orchset [5] contains 64 excerpts from symphonies, ballet suites and other musical forms interpreted by symphonic orchestras. The definition of melody in this dataset is not restricted to a single instrument, with all (four) annotators agreeing on the melody notes [15]. The focus is pitch estimation, while voicing detection is less important: the proportion of voiced and unvoiced frames is 93.7/6.3%.

3.2 Salience function evaluation

Salience functions are evaluated from two different perspectives: pitch and salience estimation accuracy. To do so, we compute four different metrics [19], using the ground truth melody.

We start by computing salience function peaks, and then select the peak closest to the ground truth, which is considered as the melody salience peak. The first metric is the frequency error of the salience function Δf_m, computed as the difference (in cents) between the frequency of the melody salience peak and the ground truth f0. The following metrics deal with salience estimation. The first metric (RR_m) is the reciprocal rank score of the melody salience peak amongst the rest of salience peaks (the closer to one the better). The second (S1) is the relative salience of the melody peak in comparison to the highest salience peak in that frame (the closer to one the better). Last metric (S3) computes the salience of the melody peak, divided by the mean salience of the 3 highest peaks (the higher the better). We consider the latter as the single most important measure, since it quantifies the ability of a method to make the melody pitch more salient than the rest of the peaks, which is a key property of a salience function.

3.3 Melody extraction evaluation

Following MIREX methodology, we evaluate melody extraction approaches by comparing the estimated sequence of pitches against a ground truth sequence of melody pitches. All evaluated algorithms were set to report an estimated melody pitch even for frames considered unvoiced. This allows evaluating voicing and pitch estimation separately. Five standard melody extraction metrics [6] are computed using mir_eval [20]: Voicing recall rate (VR): proportion of frames labelled as melody frames in the ground truth that are estimated as melody frames; Voicing false alarm rate (VFA): proportion of frames labelled as non-melody in the ground truth that are mistakenly estimated as melody frames; Raw Pitch Accuracy (RPA): proportion of melody frames in the ground truth for which the estimation is considered correct (within half a semitone of the ground truth); Raw Chroma Accuracy (RCA): measure of pitch accuracy, in which both estimated and ground truth pitches are mapped into one octave, thus ignoring octave errors; Overall Accuracy (OA): proportion of frames that were correctly labelled in terms of both pitch and voicing.

4. RESULTS

4.1 Salience function

In order to have an idea of the variance between excerpts, we compute the mean value of the metrics for each excerpt, and we then visualise evaluation results with a boxplot, as presented in Figure 5. The lower and upper lines of each

[5] mtg.upf.edu/download/datasets/orchset

[6] http://www.music-ir.org/mirex/wiki/2014:Audio_Melody_Extraction

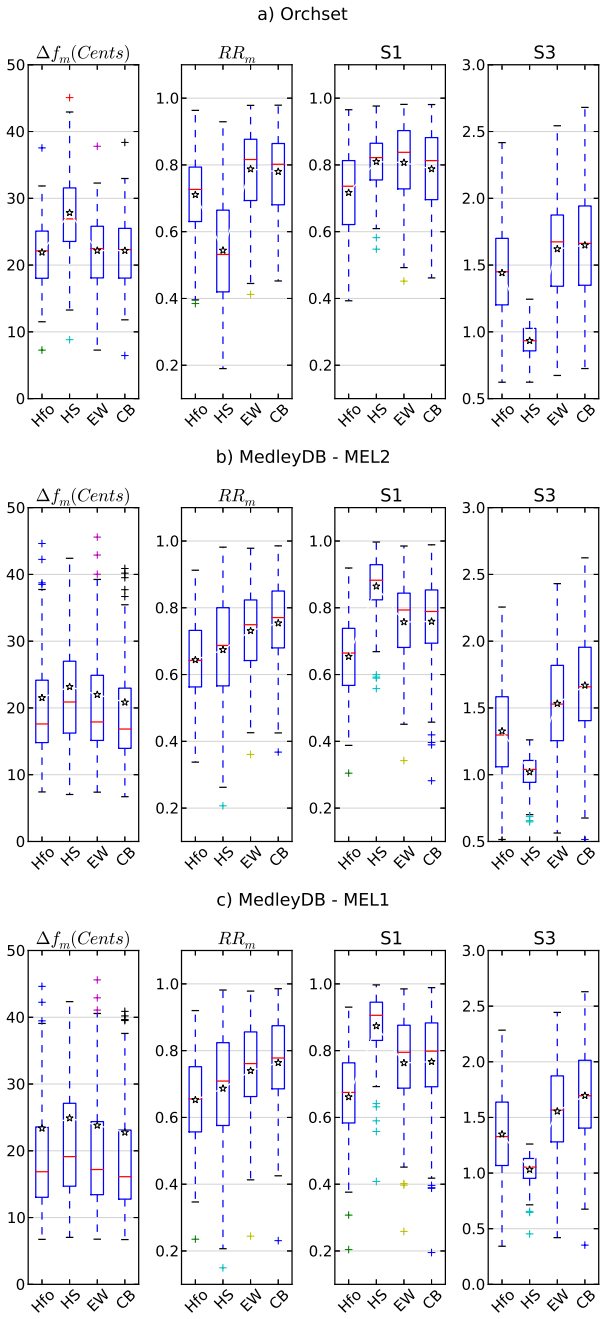


Figure 5: Salience function evaluation results a) Orchset. b) MedleyDB with MEL2 definition c) MedleyDB with MEL1 definition. Mean values represented with a star.

box show the 25th and 75th percentiles of the sample, and the line inside each box represents the median.

Before analysing results, note that the normalisations and energy weighting performed in the proposed EW salience function, do not affect any of the metrics for salience function evaluation. Any difference in results between EW and H_{f_0} is thus only due to the proposed Gaussian filtering performed in each frame of the salience function.

Regarding frequency error (Δf_m), the lowest median value is obtained with CB, but differences amongst all approaches are not significant on MedleyDB (with both melody definitions). In the case of Orchset, H_{f_0} and the proposed methods obtain lower errors than HS. Note that on Orchset, results do not really represent the difference from the closest salience peak and the real melody pitch, since melody notes are played by orchestral sections, and individual instruments contributing to the melody are playing slightly different pitches. Additionally, ground truth pitches in Orchset are actually quantized at the semitone level, since they were derived from MIDI notes, without tuning information.

With regard to salience related metrics, we observe that the reciprocal rank RR_m of the proposed salience functions EW and CB is higher than the rest. Also note that HS performs better on MedleyDB than on Orchset, while H_{f_0} behaves similarly in both datasets. The performance of CB is better than EW on MedleyDB, presumably because of the synergy obtained when combining the two salience functions. In the case of Orchset, the performance of CB in comparison to EW is decreased since HS does not perform as well in orchestral data.

HS obtains the highest mean value of S1 for MEL2 on MedleyDB, however best S3 results are obtained with CB. As previously introduced, S1 compares the salience of the melody peak and the highest salience peak in a frame. S3 measures if the melody peak stands out from the other peaks of the salience function and by how much. These results show that HS achieves a high S1 score because the highest salience peaks do not actually present a high difference between them (the value of both S1 and S3 are close to one). HS obtains a median S3 of less than 1 on Orchset, which attending to the definition of the metric, means that (in average) the salience of the melody peak is smaller than the mean of the three highest peaks. H_{f_0} on the other hand presents a higher difference between the melody peak and the following most salient peaks.

We thus conclude that the proposed combinations do not significantly reduce the estimation error of the melody pitch frequency (Δf_m) with respect to the compared approaches. However, the proposed combined salience function (CB) achieves the highest S3 value, meaning that is the most able to make the melody pitch more salient.

4.2 Melody extraction

After analysing salience functions results, we focus on complete melody extraction methods. Figure 6 shows the results for all metrics obtained with all approaches in both Orchset and MedleyDB with both melody definitions. Results are reported for experiments conducted with the same parameters values as in [3], except for the maximum allowed gap tc = 50 ms. An analysis of the effect of the parameters is presented in Section 4.3.

Comparing the results obtained with the proposed methods, we observe that CBM achieves the best overall accuracy in both datasets. This is specially noticeable in Orchset, partially due to the higher recall. Pitch related accuracies are quite similar in both approaches, especially for MedleyDB. In comparison with other methods, both of the proposed methods yield a higher OA than ESS (baseline), for both datasets and both melody definitions. The OA is also higher in comparison to the rest of the related approaches, for both MEL1 and MEL2 on MedleyDB. In

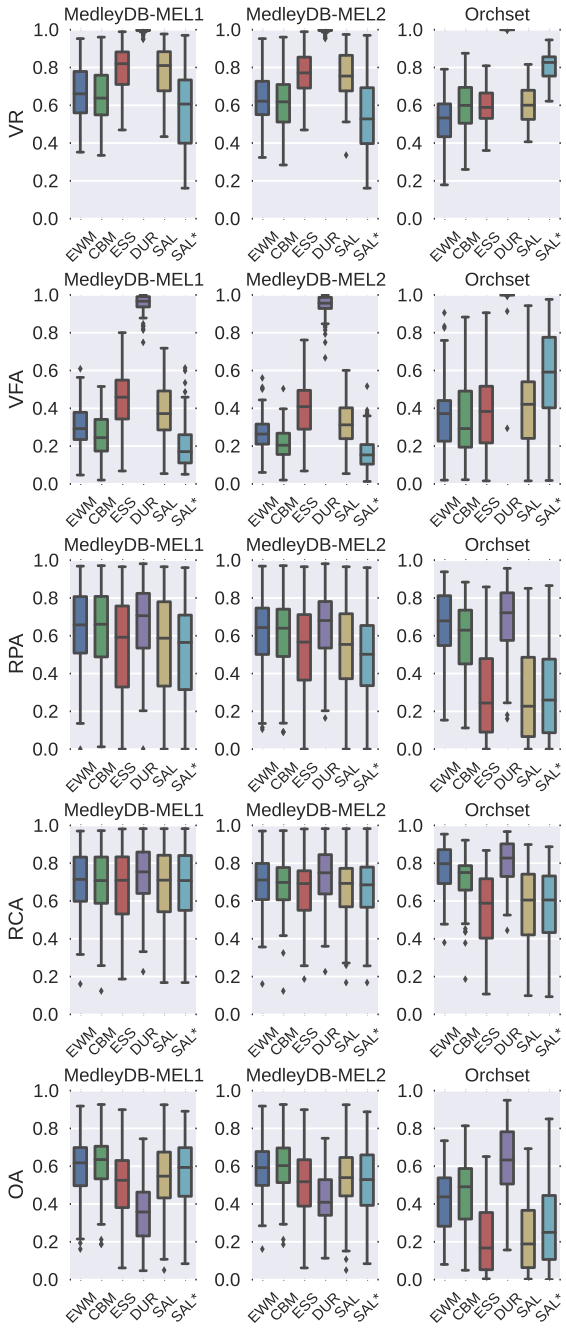


Figure 6: Evaluation results on all metrics, for MedleyDB with both MEL1 and MEL2 definitions and Orchset. "SAL*" denotes the results obtained with SAL with $\nu = -1$ for MedleyDB and $\nu = 1.4$ for Orchset

the case of Orchset, only DUR yields a better OA than the proposed methods, due to a very high recall. Note that DUR always obtains almost perfect recall on all datasets, and very high false alarm rates, since this method outputs almost all frames as voiced. The influence of this fact on the overall accuracy depends on the amount of voiced frames of the dataset. Since Orchset mostly contains voiced frames (93.7%), it is beneficial, but MedleyDB contains full songs with large unvoiced portions, and false alarms in this data considerably reduce the OA.

The proposed approaches achieve a slightly lower RPA in comparison to H_{f_0}. This is related to the fact that Durrieu's method estimates most frames as voiced. Even though it is considered a pitch related metric, RPA is actually also affected by voicing estimation, since it compares estimated pitches with voiced ground truth pitches. If some of the melody contours are not created, or erroneously filtered (e.g. due to a lower salience in comparison to the rest of the contours), this will affect both voicing related metrics and pitch related metrics. That is the case for our proposed methods: while many frames are correctly identified as unvoiced, some contours which correspond to the melody are filtered or simply not created, which decreases pitch related accuracies. However, reducing the voicing false alarm rate helps achieving a better overall accuracy.

SAL and ESS obtain lower pitch related accuracies (RPA, RCA) than the proposed methods, specially in orchestral music. Given that the only difference between them is the salience function, we can conclude that the results are improved thanks to the use of a source-filter model and gaussian filtering. This could be expected from the previously presented salience function evaluation results, since the proposed salience functions are able to make the melody pitch more salient.

Also note that the difference between RCA and RPA is much higher in SAL than in the proposed methods, specially on Orchset. This shows that the kind of signal representation underneath the proposed pitch salience functions is very effective at reducing the amount of octave errors [6, 21].

4.3 Parameter tuning

Previous results can be further improved by adapting melody extraction parameters to the proposed salience functions. We first analysed the influence of Gaussian filtering (see Figure 2) on the complete melody extraction system CBM, by suppressing it from the pitch salience creation process. The effect is quite small on MedleyDB, but it helped improving pitch estimation on Orchset (4% points). This could be due to the small differences in the pitch played by the individual instruments contributing to the melody. As previously observed with the salience function evaluation results, by smoothing H_{f_0} we are able to make more salient the pitches of the notes played by orchestral sections in unison.

Several other parameters affect different parts of the method: salience function creation, contour creation or melody contour selection. Figure 7 shows the effect of the number of iterations (N_{iter}), maximum allowed gap in the contour ($tc \in \{50, 75, 100\}$ ms) and voicing tolerance parameter ($\nu \in \{-1, 0.2, 1, 1.4\}$). For the sake of clarity, we only show results from CBM, since the highest overall accuracy was obtained with this method. Results obtained with other methods are also presented, including the effect of N_{iter} on DUR. Best results in vocal music are obtained with few iterations, but complex data (such as instrumental, and especially orchestral music) benefits from a higher number of iterations.

In any case, the influence of pitch salience creation parameters is relatively small in comparison to the influence

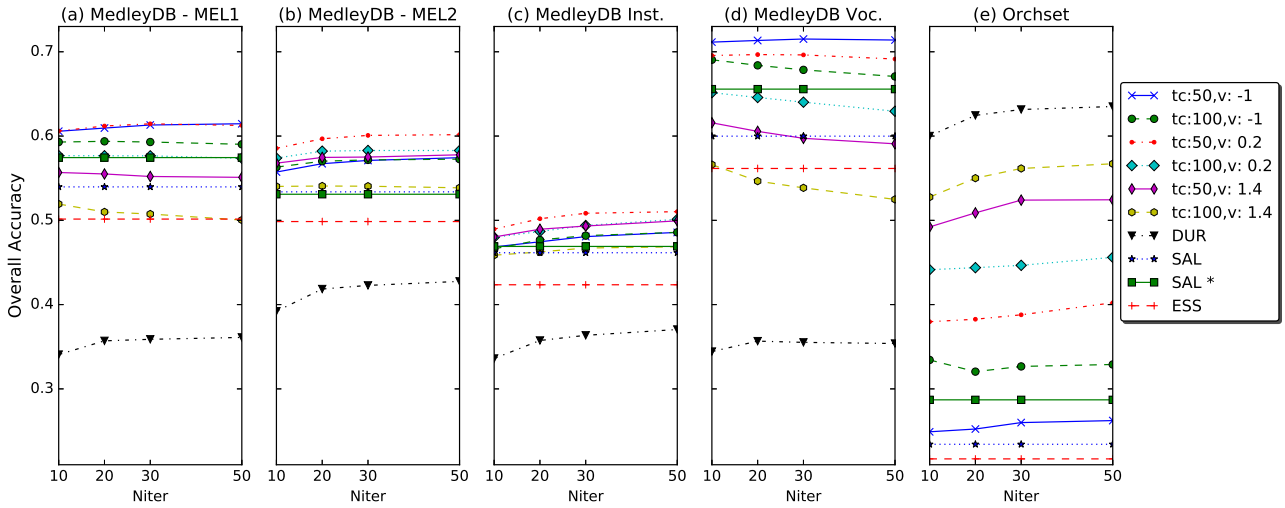


Figure 7: Overall accuracy vs. number of iterations, and influence of time continuity (tc), and voicing (ν) parameters on the results obtained with CBM. Results for DUR, SAL and the baseline (ESS) given as a reference. "SAL*" denotes that $\nu = -1$ for MedleyDB and $\nu = 1.4$ for Orchset (a) MedleyDB (MEL1) definition; (b) MedleyDB (MEL2); (c) MedleyDB (MEL1), instrumental songs; (d) MedleyDB (MEL1), vocal songs; (e) Orchset.

of pitch contour tracking parameters. For instance, OA generally increases considerably when the maximum gap between pitches in a contour is decreased from 100 ms to 50 ms. This is probably due to the noise added in unvoiced frames by the SIMM, which can partially be filtered in the contour creation process. The effect of the voicing parameter (ν) is evident: a higher value increases the voicing threshold and less contours are filtered, which is beneficial in Orchset. Setting a lower threshold is benefitial in MedleyDB with MEL1 definition, since the amount of voiced frames is smaller. Default peak filtering parameter values (τ_σ, τ_+) provided good results in MedleyDB, but OA can be increased up to 60% in Orchset, by increasing τ_σ from 0.9 to 1.3 with $\nu = 1.4$. This allows a higher difference in salience below the salience mean during pitch contour creation, which is appropriate to deal with the higher dynamic range in classical music.

Regarding instrumentation, OA in MedleyDB vocal music is higher than in instrumental, but with the proposed method, we increased it in about 10 and 8 percentage points (pp) over the baseline (ESS) respectively. The improvement is even more evident in Orchset. According to the results, we can conclude that our salience function leads to a better accuracy than HS, for both single instruments and instrument sections.

CBM obtained 25 percentage points (pp) higher OA in MedleyDB (with MEL1 definition, see Figure 7) compared to DUR, and slightly worse in Orchset (around 4 pp with the best parameters mentioned. Additionally, CBM generally needs less iterations (N_{iter}) compared to DUR to achieve the best results, which is very positive given the high computational weight of the estimation algorithm. In comparison to the approach by Salamon et al., we obtained 5 and 30 pp higher accuracy in MedleyDB (with MEL1 definition) and Orchset respectively, using the best voicing parameter for each dataset in both algorithms (CBM and SAL*). This corresponds to about 10% and 100% relative increase, due to the low accuracy of SAL in Orchset.

The selection of parameters has been here performed automatically, but it could be performed automatically by selecting the best performing configuration in a training set. Another possibility is to use a pitch contour classification approach [10], by training a classifier to distinguish between melody and non-melody pitch contours using the proposed salience functions [22].

5. CONCLUSIONS

This paper presents a melody extraction method based on the combination of a source-filter model and pitch contour based tracking. We proposed two different salience functions and we have shown that Gaussian filtering and the combination of a source-filter model with harmonic summation help increasing the salience of melody pitches. The signal representation employed proved to improve pitch estimation accuracy and to reduce octave errors in comparison to harmonic summation. Our complete melody extraction method obtains similar or higher overall accuracy in comparison to similar approaches, when evaluated on a large and varied dataset. This is achieved by accurate voicing detection and pitch estimation.

Future work deals with improving the salience function, in order to further reduce the amount of noise in unvoiced parts, and to improve the adaptation to the contour creation process. We also foresee the use of a supervised method for pitch contour classification and melody tracking.

Acknowledgments

This work is partially supported by the European Union under the PHENICX project (FP7-ICT-601166) and the Spanish Ministry of Economy and Competitiveness under CASAS project (TIN2015-70816-R) and Maria de Maeztu Units of Excellence Programme (MDM-2015-0502).

6. REFERENCES

[1] J. Salamon, E. Gómez, D. Ellis, and G. Richard, "Melody Extraction from Polyphonic Music Signals: Approaches, applications, and challenges," *IEEE Signal Process. Mag.*, vol. 31, pp. 118–134, 2014.

[2] A. Klapuri, "Multiple fundamental frequency estimation by summing harmonic amplitudes," in *Proc. ISMIR*, 2006, pp. 216–221.

[3] J. Salamon and E. Gómez, "Melody extraction from polyphonic music signals using pitch contour characteristics," *IEEE Trans. Audio. Speech. Lang. Processing*, vol. 20, no. 6, pp. 1759–1770, 2012.

[4] K. Dressler, "Towards Computational Auditory Scene Analysis: Melody Extraction from Polyphonic Music," in *Proc. CMMR*, 2012.

[5] ——, "Multiple fundamental frequency extraction for MIREX 2012," in *Music Inf. Retr. Eval. Exch.*, 2012.

[6] J. Durrieu, B. David, and G. Richard, "A musically motivated mid-level representation for pitch estimation and musical audio source separation," *Sel. Top. Signal Process. IEEE J.*, vol. 5, no. 6, pp. 1180–1191, 2011.

[7] B. Fuentes, A. Liutkus, R. Badeau, and G. Richard, "Probabilistic model for main melody extraction using constant-Q transform," in *Proc. IEEE Int. Conf. on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2012, pp. 5357–5360.

[8] J. Durrieu, G. Richard, B. David, and C. Févotte, "Source/filter model for unsupervised main melody extraction from polyphonic audio signals," *Audio, Speech, Lang. Process. IEEE Trans.*, vol. 18, no. 3, pp. 564–575, 2010.

[9] J. Salamon, G. Peeters, and A. Röbel, "Statistical characterisation of melodic pitch contours and its application for melody extraction," in *13th Int. Soc. for Music Info. Retrieval Conf.*, Porto, Portugal, Oct. 2012, pp. 187–192.

[10] R. Bittner, J. Salamon, S. Essid, and J. Bello, "Melody extraction by contour classification," in *Proc. International Society of Music Information Retrieval (ISMIR)*, October 2015.

[11] S. Ternström and J. Sundberg, "Intonation precision of choir singers," *The Journal of the Acoustical Society of America*, vol. 84, no. 1, pp. 59–69, 1988.

[12] R. Bittner, J. Salamon, M. Tierney, M. Mauch, C. Cannam, and J. Bello, "Medleydb: a multitrack dataset for annotation-intensive mir research," in *Proc. ISMIR*, 2014, pp. 155–160.

[13] J. Bosch, R. Marxer, and E. Gómez, "Evaluation and combination of pitch estimation methods for melody extraction in symphonic classical music," *Journal of New Music Research*, DOI: 10.1080/09298215.2016.1182191, 2016.

[14] J. S. Downie, "The music information retrieval evaluation exchange (2005-2007): A window into music information retrieval research," *Acoustical Science and Technology*, vol. 29, no. 4, pp. 247–255, 2008.

[15] J. Bosch and E. Gómez, "Melody extraction in symphonic classical music: a comparative study of mutual agreement between humans and algorithms," in *Proc. 9th Conference on Interdisciplinary Musicology – CIM14*, Berlin, 2014.

[16] C. Hsu and J. Jang, "Singing pitch extraction by voice vibrato/tremolo estimation and instrument partial deletion," in *Proc. ISMIR*, 2010, pp. 525–530.

[17] T. Yeh, M. Wu, J. Jang, W. Chang, and I. Liao, "A hybrid approach to singing pitch extraction based on trend estimation and hidden markov models," in *Proc. IEEE Int. Conf. on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2012, pp. 457–460.

[18] D. Bogdanov, N. Wack, E. Gómez, S. Gulati, P. Herrera, O. Mayor, G. Roma, J. Salamon, J. Zapata, and X. Serra, "Essentia: an open source library for audio analysis," *ACM SIGMM Records*, vol. 6, 2014.

[19] J. Salamon, E. Gómez, and J. Bonada, "Sinusoid extraction and salience function design for predominant melody estimation," in *Proc. 14th Int. Conf. Digit. Audio Eff. (DAFx-11), Paris, Fr.*, 2011, pp. 73–80.

[20] C. Raffel, B. McFee, E. Humphrey, J. Salamon, O. Nieto, D. Liang, and D. Ellis, "mir eval: A transparent implementation of common mir metrics," in *Proc. ISMIR*, 2014.

[21] M. Goto, "A Real-Time Music-Scene-Description System: Predominant-F0 Estimation for Detecting Melody and Bass Lines in Real-World Audio Signals," *Speech Communication*, vol. 43, no. 4, pp. 311–329, Sep. 2004.

[22] J. Bosch, R. M. Bittner, J. Salamon, and E. Gómez, "A comparison of melody extraction methods based on source-filter modelling," in *Proc. ISMIR*, New York, Aug. 2016.

SOUNDSCAVENGER: AN INTERACTIVE SOUNDWALK

Naithan Bosse
School of Creative and Performing Arts
University of Calgary
naithanbosse@ucalgary.ca

ABSTRACT

SoundScavenger is an open-form, networked soundwalk composition for iOS devices. Embedded GPS sensors are used to track user positions within a series of regions. Each region is associated with a different series of soundfiles. The application supports networked interactions, allowing multiple users to explore and communicate within a semi-shared soundscape.

1. INTRODUCTION

SoundScavenger divides a geographic region of approximately 200 square meters into 7 distinct zones (figure 1). Each zone is associated with a different soundscape. The GPS sensors embedded in the iOS devices allow the SoundScavenger application to play the correct soundfiles as the user wanders from one zone to another. Using Apple's Game Center, two users can collaborate in exploring a semi-shared soundscape. The users hear soundscapes associated with the GPS coordinates of both participating users, allowing them to explore many different combinations of sounds as they move between different physical locations. Users are also able to interact with one another by using the application interface to cue sounds from a collection of acousmatic gestures.

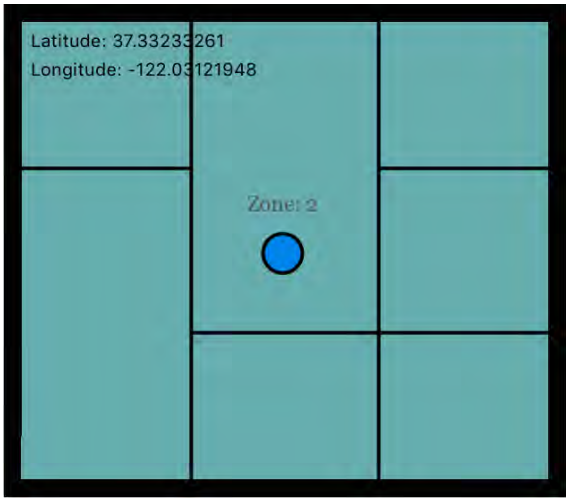


Figure 1. The user interface: A GPS map divided into 7 zones.

2. MOBILE-MUSIC

Existing mobile-music projects range from ensembles [1, 2], large-scale compositions [3], augmented reality [4] and digital controllers [5]. In "Updating the Classifications of Mobile-Music Projects", David John identifies location and collaborative composition as two recurring themes in current mobile-music projects [6]. These categories are of primary importance to SoundScavenger. SoundScavenger aims to continue in the example set by projects such as *AuRal* and *Net_Dérive* [7, 8], and is particularly inspired by the soundwalks of Janet Cardiff [9].

David Kim-Boyle argues that networked projects "resituate the role of the composer to that of designer and transform the nature of performance to that of play." [10] Listening to a composition becomes an act of engaged experimentation rather than a passive experience. Many mobile-music projects, such as *Biophilia* or *Polyfauna,* use this same approach of designing musical environments rather than composing strictly scored material [11, 12]. SoundScavenger also adopts this approach, aiming to provide the user with some amount of agency within the musical experience.

3. INTERACTION DESIGN

Upon starting the app, users are placed on a map separated into 7 zones, illustrating the locations of the 7 soundscapes. Through this map, the user is made aware of the existence of locational boundaries. In order to help the user understand the relationship between their physical position and the resulting soundscape, adjacent zones often contrast one another in terms of sound material. By using timbre to create a clear delineation between zones, the user can more quickly understand an available form of interaction. It is especially important to clarify this form of interaction if the user begins a multi-player session. A multi-player session overlaps the audio of two zones at any given time. Transitions between zones may be somewhat obscured by the overlap.

Initial users of the system have actively explored the locational boundaries by to-ing and fro-ing between zones in order to test the correlation between the boundaries drawn on the map and the resulting change of sonic material. Several crossfade times between 10 milliseconds and 20 seconds were tested to smooth the transitions between zones. Shorter crossfades allowed the user interaction to be perceived more clearly while longer

crossfades helped to smooth jitter in the GPS input. A 2-second crossfade was selected as a satisfying compromise.

The delineation between zones is also reinforced by the touch-based controls. Upon pressing any zone on the map, a short 1-5 second gesture will be performed (figure 2). The touch controls divide the material into families of sound objects. By experimenting with the touch controls, users will hopefully connect the families of gestures with the various sound objects used to compose the 7 soundscapes. By learning the individual gestures, the user will also hopefully learn to identify when a remote user cues a gesture.

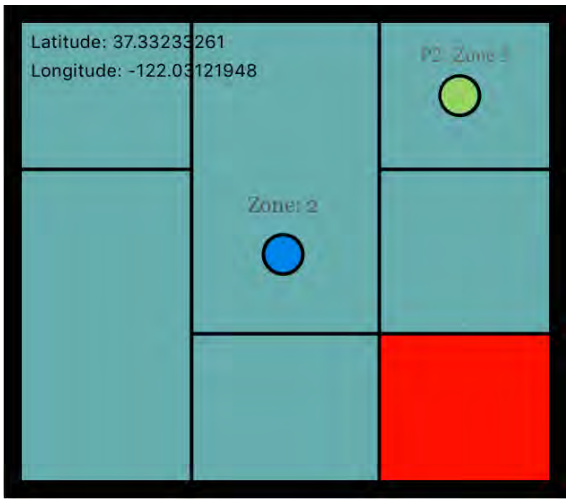


Figure 2. The user interface. The remote user is displayed as a green circle. Zone 7 is being pressed by the local user which will cause a short soundfile to play for both users.

4. A SONIC FLASHLIGHT

While the moment to moment character of SoundScavenger is nonlinear and dependent on the decisions of the listener, the large-scale formal development of the composition is primarily linear. The soundfile associated with any individual zone could be experienced on its own as an autonomous, fixed composition between 10 and 14 minutes in length. While the user is able to switch between zones at any point in time, the zones are all composed with a parallel linear trajectory. The playheads for all zones continue forward in time regardless of which zone the user is presently hearing. I perceive this approach as a 3-dimensional musical form. While the first 2 dimensions are dependent on user input, the 3rd dimension, time, is fixed, resulting in a predetermined musical arc (figure 3). The role of the user in shaping the composition is similar to a person shining a flashlight on fragments of a painting in a dark room. While the painting as a whole possesses an overall coherence, the spectator only experiences a fraction of the painting at a time. By focusing on only a restricted section of SoundScavenger at a time, the listener is able to perceive details that may be lost when listening to the work in its entirety.

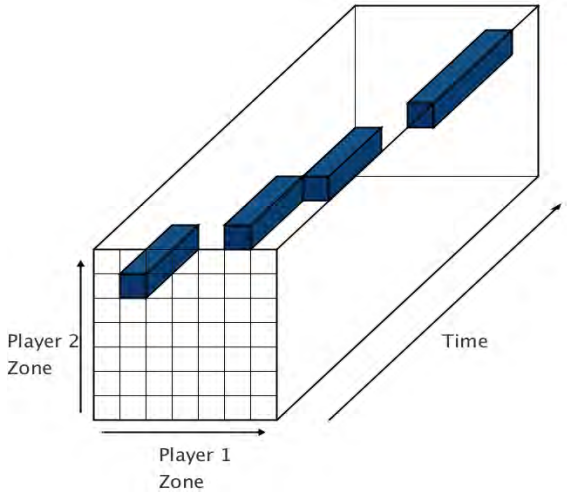


Figure 3. Mapping the player movements over time. While the music progresses linearly in time, the performers help determine the moment to moment soundscape. In this example Player 1 begins in zone 2 and moves to zone 3, 5, and 6. Player 2 begins in zone 2, moves to zone 3, and returns to zone 2.

5. SITE-SPECIFIC SOUND-DESIGN

Given the mobile nature of this composition, it is likely that the soundscapes associated with the listening locations will be noisier than a typical concert hall or recording studio. While this would generally be considered be an unfortunate quality for most pieces of electroacoustic music, (A “lo-fi” listening environment limits the audible dynamic range and may mask frequencies), prior knowledge of the listening environment allows the composer to choose sound materials for the location itself. Selecting sounds for a specific location can increase immersion by creating an ambiguous relationship between the sounds coming through the headphones and the external soundscape. This relationship is particularly effective when orchestrating for an active or busy soundscape such as a mall. For example, in a crowded mall, quickly approaching footsteps on tile may cause the listener to turn around to check whether the sound is within the composition itself or an actual person approaching. Yelling, laughing, or crying may also be more effective in environments where these sounds are natural occurrences. This effect is emphasized in SoundScavenger using binaural spatialization to place sounds outside of the visual field. Binaural spatialization was accomplished in Max using an FFT-based binaural panning tool by Jakob Andersen [13].

Rather than using sounds which are native to the listening environment, In his sound installation *Auditory Tactics*, Gauthier takes the opposite approach, using close vocal sounds to contrast the “public sphere” with sounds more generally associated with an intimate environment [14].

Adopting sounds which are perceived as close to the head or out of place within the context of the listening environment and contrasting these sounds with the soundscape of the listening environment creates a novel sonic experience. Janet Cardiff also uses this technique of contrasting the public and private spheres in many of her soundwalks, often pairing soft, close mic'd vocals with soundscapes of the surrounding environment [9].

SoundScavenger is originally designed for use on the University of Calgary campus. However, the application does not limit the user to a specific location. The GPS coordinates are instead based on the user's location at the moment the app is launched. Although the site-specific sound-design is compromised when used outside of the originally intended location, many of these techniques may still be translated to similar locations. For example, a work designed for use in a library would likely translate well to different libraries of a similar size. Selecting intimate sounds in order to contrast a public location with a private soundscape would likely remain effective in any sufficiently public setting.

Wrightson argues that many people try to avoid or ignore noisy soundscapes by using sound-proofing or adding "acoustic perfume-music":

> Music – the virtual soundscape – is, in this context, used as a means to control the sonic environment rather than as a natural expression of it [15].

Site-specific sound-design could be used not to control or ignore a soundscape, but instead to engage with it directly, integrating it as an element in the composition itself. Hopefully, this approach to composition could lead to a deeper appreciation of the external soundscape.

6. FUTURE WORK

Moving forward, SoundScavenger will be extended to improve both the depth of immersion and the range of interaction. By implementing tools for sound-analysis, the application could play specific gestures or soundfiles mimicking or contrasting a previously unknown soundscape (for example, performing traffic sounds in environments which feature high levels of continuous broadband noise). SoundScavenger could enhance the interaction and localization between players by implementing live spatialization, associating the remote player with both a specific soundscape and location. To be effective, the application would use the iOS device's built-in sensors to ascertain the user's orientation with regard to the remote player. The sounds associated with the remote player would emanate from the appropriate direction regardless of the local player's orientation. Finally, it may be effective to use live filtering to create a sense of distance associated with specific soundfiles, so that as a user nears a specific location, the sounds are perceived as gradually getting closer.

7. CONCLUSIONS

The software implementation of SoundScavenger functions as an interactive stage for the composition itself. Many different types of music can be performed on this stage. However, each GPS soundwalk will need to consider its use of acoustic environment, interaction design, and musical form.

SoundScavenger is freely available as an iOS app on the iTunes App Store. An exported, model version of SoundScavenger is also available at `www.naithan.com/soundscavenger`.

Acknowledgments

SoundScavenger is developed with support by the Social Sciences and Humanities Research Council of Canada.

8. REFERENCES

[1] G. Wang, G Essl, and H. Penttinen, "Do Mobile-Phones Dream of Electric Orchestras?" in Proc. International Computer Music Conference (ICMC2008), Belfast, 2008.

[2] N. d'Allesandro, A. Pon, J. Wang, D. Eagle, E. Sharlin, and S. Fels, "A Digital Mobile Choir: Joining Two Interfaces towards Composing and Performing Collaborative Mobile Music," in Proc. Int. Conf. New Interfaces for Musical Expression (NIME2012), Ann Arbor, Michigan, 2012.

[3] G. Levin, S. Gibbons, and G. Shakar, "Dialtones: A Telesymphony," 2001. Accessed April 27, 2016. `http://www.flong.com/storage/experience/telesymphony`

[4] A. Schianchi, "Sin título (site specific ubicuity)," Schianchi.com, 2011, Accessed April 28, 2016. `http://schianchi.com.ar/obras/sintitulo2011.html`

[5] Hexler, "TouchOSC," Accessed April 27, 2016. `http://www.hexler.net`

[6] J. Allison and C. Dell, "AuRal: A Mobile Interactive System for Geo-Locative Audio Synthesis," in Proc. Int. Conf. New Interfaces for Musical Expression (NIME2012), Ann Arbor, Michigan, 2012.

[7] A. Tanaka and P. Gemeinboeck, "Net_Dérive," 2006. Accessed October 19, 2014.

[8] J. Cardiff, "Walks," Accessed April 27, 2016. `http://www.cardiffmiller.com/artworks/walks`

[9] D. John, "Updating the Classifications of Mobile Music Projects," in Proc. Int. Conf. New Interfaces for Musical Expression (NIME2013), DaeJeon, Republic of Korea, 2013, pp. 301-306.

[10] D. Kim-Boyle, “Network Musics: Play, Engagement and the Democratization of Performance,” *Contemporary Music Review* 28, no. 4, pp. 363-375.

[11] Bjork and S. Snibbe, “Biophilia,” App Store, 2011. Accessed April 27 2016. https://itunes.apple.com/ca/app/bjork-biophilia

[12] T. Yorke, N. Godrich, S. Donwood, and Universal Everything, “PolyFauna,” App Store, 2014. Accessed April 27, 2016. https://itunes.apple.com/ca/app/polyfauna

[13] J. Andersen, “FFT-based Binaural Panner,” *Jakobhandersen*, 2011, Accessed July 17th 2016 http://jakobhandersen.dk/projects/fft-based-binaural-panner

[14] P. Gauthier and P. Pasquier, “Auditory Tactics: A Sound Installation in Public Space Using Beamforming Technology,” *Leonardo* 43 no. 5, MIT Press, 2010, pp. 426-433.

[15] K. Wrightson, “An Introduction to Acoustic Ecology,” *eContact* 5.3, Accessed April 27, 2016. http://econtact.ca

SOUND FOREST/LJUDSKOGEN: A LARGE-SCALE STRING-BASED INTERACTIVE MUSICAL INSTRUMENT

Roberto Bresin, Ludvig Elblaus
Emma Frid, Federico Favero
KTH Royal Institute of Technology
{roberto, elblaus
emmafrid, ffavero}
@kth.se

Lars Annersten
David Berner
Musikverket
{lars.annersten,
david.berner}
@musikverket.se

Fabio Morreale
Queen Mary University of London
f.morreale
@qmul.ac.uk

ABSTRACT

In this paper we present a string-based, interactive, large-scale installation for a new museum dedicated to performing arts, Scenkonstmuseet [1], which will be inaugurated in 2017 in Stockholm, Sweden. The installation will occupy an entire room that measures 10x5 meters. We aim to create a digital musical instrument (DMI) that facilitates intuitive musical interaction, thereby enabling visitors to quickly start creating music either alone or together. The interface should be able to serve as a pedagogical tool; visitors should be able to learn about concepts related to music and music making by interacting with the DMI. Since the lifespan of the installation will be approximately five years, one main concern is to create an experience that will encourage visitors to return to the museum for continued instrument exploration. In other words, the DMI should be designed to facilitate long-term engagement. Finally, an important aspect in the design of the installation is that the DMI should be accessible and provide a rich experience for all museum visitors, regardless of age or abilities.

1. INTRODUCTION

The realization of interactive installations in museums and science centers has become increasingly popular throughout the past two decades [1, 2]. Such installations may engage visitors, provide rewarding experiences that stimulate learning and motivate visitors to return to the museum. In this paper we present a new installation for a new museum dedicated to performing arts, Scenkonstmuseet [1], Swedish Museum of Performing Arts, which will be inaugurated in 2017 in Stockholm, Sweden. The museum will be organized in three sections: Dance, Theater and Music. The installation presented in this paper will be part of the Music section and will consist of a large-scale digital musical instrument (DMI) [3]. The DMI will occupy an entire room which measures 10x5 meters. The installation is the result of collaboration between the Swedish Museum of Performing Arts and KTH Royal Institute of Technology in Stockholm, Sweden. In the following section we briefly describe the theoretical framework that has served as foundation for our design decisions, followed up by a detailed description of the Sound Forest installation.

[1] http://www.scenkonstmuseet.se

2. BACKGROUND

2.1 Collaborative Musical Interface Design for Novice Players

The Sound Forest installation should not only provide a rich musical experience for the single novice player but also enable collaborative musical experiences. When designing collaborative experiences for novice players in public settings, one should strive to achieve a balance between simplicity and virtuosity while at the same time minimize the time required to learn how to use the interface [4]. The trade-off in determining the appropriate balance of complexity and expressivity of a musical interface is not easily resolved [5]. Designers of interfaces for musical expression for public settings usually address the need to cater to novice users by restricting the musical control. It has been suggested that providing novices with easily accessible music making experiences is more important than providing complex interfaces with upward capability for virtuosic expression [5] in this context. As discussed in [6], limiting the number of features in the design of a musical interface can be beneficial for the performer.

Nevertheless, in order to encourage long-term engagement, the initial ease of use should be coupled with a long-term potential for expansion to virtuosity, as suggested in [7]. In general, activities which remain engaging in the long term are often characterized by a trade-off between ease of learning and long-term power and flexibility [8]. Engaging, flow-like [9] activities such as music are characterized by being at an appropriate level of difficulty [10]. The interaction in the Sound Forest should be designed in such a manner to avoid "dead ends" [10]; the complexity of the instrument should provide a possibility for unlimited growth and encouragement.

2.2 Accessible Interactive Musical Interface Design

The underlying premise of most collaborative interface design is that playing music can be made accessible to non-

musicians through the use of various design constraints [5]. The term accessibility in this context does however not only involve designing for novices, but also for people with impairments. There are numerous examples of research exploring approaches for customizing musical interfaces to people with impairments (see e.g. [11–16]). There are also examples of "accessible" music interfaces, such as e.g. *Skoog* [2] and *Soundbeam* [3].

It is important to note that an inclusive design approach is preferable when designing for sensory impaired; design issues should considered at the beginning of the design process, so that the design is done for the visitors abilities, rather than compensating for their disabilities [17]. It has been found that children with learning disabilities were able to do their best when presented with learning and creating music in a multisensory learning environment and that "the better functioning modes of learning helped the child to compensate for the dysfunctioning modes" [18]. The Sound Forest will provide the player with multiple modes of interaction. This multimodal property of the room, in which sound, visual and haptic feedback will be provided, is one aspect that we believe will lead to inclusion of different visitor groups.

3. THE SOUND FOREST - LJUDSKOGEN

Some design requirements and constraints were defined by the curators of the museum in the initial stage of the development process of the interactive installation. The installation should be designed in such a manner that it enables visitors without any prior knowledge of musical instruments to engage with the DMI in a rewarding way. A key concern is to create a digital instrument facilitating intuitive musical interaction enabling visitors to quickly start creating music either alone or together. The interface should also be able to serve as a pedagogical tool; visitors should be able to learn about concepts related to music and music making by interacting with the DMI. Since the installation will be set at the museum for a period of five years, one important aspect is to create an experience that will encourage visitors to return to the museum to continue to explore the instrument. The design of the installation shall focus on sustaining long-term engagement with the system. Finally, an important aspect of the installation is accessibility; ensuring that the installation is not only easily accessible for people with impairments (e.g. blind, deaf or visitors with impaired mobility) but also able to provide a rich experience for these visitors. The room itself should not create barriers that hinder persons with impairments to engage in a musical experience.

After a period of about six month during which we had several discussions and brainstorming meetings about how to comply with the requests from the curators, we came up with the idea of an installation based on a string metaphor. Several researchers and artists have user the "string" as controller in different installations and new DMIs, such as the *Manipuller* [19, 20], the Web (by Michel Waisvisz [21]), the *STRIMIDILATOR* [22], the *Vocal Chorder* [23], *Global String* and *the SoundNET* [24], to name a few.

A string has clear affordances well known by most of the museum visitors and invites to different types of interaction such as plucking, bowing, punching, pulling, pushing, scraping and brushing. The central idea was to create an interactive music room that could serve as a traditional acoustic string instrument in which long strings attached to the ceiling and floor would serve as a control interface. As a metaphor of traditional acoustic string instruments, we wanted the movements and feeling when interacting with the strings to be tightly connected to the quality of the sound and the physical interaction. The main idea was that energy provided by body gestures performed by visitors when interacting with the strings would be translated into energy, affecting the acoustic properties of sound and of other perceptual modalities such as haptic feedback and lighting. The presence of sound, lights and visual effects as well as haptic feedback will support the intentions of the players and reinforce the perception of a highly responsive system. The final goal is an installation that quickly and intuitively provides the feeling of being a musician. This includes being able to play and create music both on your own as well as together with other visitors. A sketch of the final installation can be seen in Figure 1. We aimed at creating a setting inspired by a forest in which strings would serve as metaphoric trees which, with help from lightning design, would create a mystical setting (see conceptual sketch in Figure 2). The installation room was named Sound Forest (or Ljudskogen, in Swedish). In the following sections we describe the different components used for creating the interactive strings more in detail.

Our work is novel in the sense that, to the best of our knowledge, no prior large-scale multisensory installation has explored aspects of multimodal interaction in a collaborative setting involving multiple mono-cord ceiling-to-floor strings. The fact that the installation will be in place for five years will enable us to run multiple player studies involving different visitor groups (age, abilities, size, education) making it possible to investigate thousands of users, also in longitudinal studies over the 5-year period. Sound Forest will enable us to study how the room could be used for educational purpose, such as e.g. for improvising music in a group setting, or appreciating different sounds generated through different synthesis models.

4. CONCEPTUAL DESIGN AND PROTOTYPING

As briefly presented above, the requirements by the museum curators and pedagogues for The Sound Forest can be summarized as follows: the design and realization of the installation should:

- *Enable* interaction, creativity, participation, engagement
- *Foster/promote* learning, education
- *Include* different user groups

The instrument should be:

[2] http://skoogmusic.com/
[3] http://www.soundbeam.co.uk/

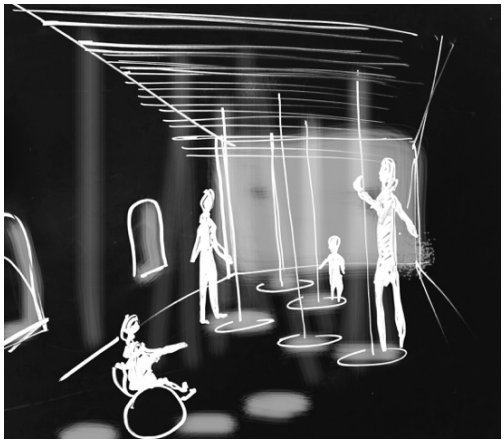


Figure 1: Sketch of the final installation with five strings, vibration platforms underneath each string, and glowing light emitted by each string.

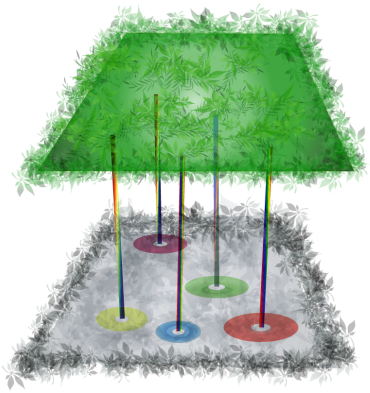


Figure 2: Conceptual sketch.

- *Scalable*: functioning for one single player as well as for many players; expandable in order to enable creative development
- *Intuitive*: clear affordances
- *Collaborative*: allow interaction of several people at the same time
- *Accessible*: visitors should be equally able to use the instrument, regardless of age and/or abilities
- *Easy to maintain*: about 12 000 visitors per month will be visiting the museum

These specifications will be met also by taking into account the tight connection between sound and music and motion as emerged from several research projects during the last 20 years [25,26]. Specifically, it is not possible to generate sounds with an instrument without engaging in some form of body motion that injects energy into the instrument.

4.1 Design of One String

The instrument that we envision in this project has a mono-cord string as its basic element and metaphor. Strings provide affordances that do not need explanations; they allow for intuitive and immediate use. In order to realize the objectives of the project we will create augmented strings which allow interaction, creativity, participation, and engagement. The mono-cord strings will have the following characteristics:

- String material: LED light intertwined fiber optic cable with DMX controller
- Sensors detecting the string movements and vibrations
- Sensors detecting hand position on the string
- Sound generation: the sound will be provided through a directional loudspeaker positioned on the top of each string
- Haptic floor: a vibrating platform placed below each string that will be activated through interaction with the string itself

4.1.1 String Sensors

The string installation will, from a conceptual perspective, be divided into two parts: the installation and the collection of content. The installation comprises all the hardware and software that is needed to provide feedback, such as light, sound, and vibration, and gather data from the interaction. The collection of content can be seen as a repertoire, e.g., a set of musical works, études, pedagogical examples, perceptual experiments, that can be loaded into and performed in the installation. Creating the content will be an evolving long-term process that will include commissioning pieces by composers, inviting students to create experiments, and prototyping new forms of interaction as a component in interaction design and sound and music computing research. The project ambition is to be able to deliver more than a fixed piece and instead aim to provide a platform for further development, i.e. an instrument, rather than a fixed installation.

To support such wide-ranging activities, a broad strategy for data gathering is adopted. Each string will be fitted with a set of sensors: a high quality, full bandwidth, contact microphone; a high resolution accelerometer, at the top end of the string; and ultrasonic distance sensing, possibly both from the ceiling and the floor, depending on the emission angle of the sensor used. The possibility to weave custom sensing materials into the strings themselves, allowing for e.g. capacitive sensing, is also currently being explored. These sensors will be connected to appropriate control and capture hardware, made up of single-board computers such as Arduino or the Raspberry Pi, fitted with suitable analog to digital converters, voltage dividers, or other necessary circuitry.

All of the data, from all of the sensors, will be collected and made available for the content creators in a unified form as Open Sound Control-formatted data. In addition to the raw data, some high level features will also be computed, such as level of activity in the room, to aid content creators to interpret the wealth of data that the installation will produce.

The Sound Forest will be organized as a synchronous (real-time) centralized network [27], allowing players to interact through strings that do not have direct influence on each other. Data from players will be sent to a computerized hub for analysis and generation of musical output.

4.1.2 Haptic Floor

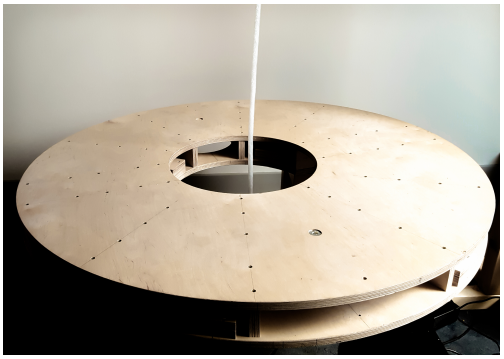
The potential of integrating vibrotactile feedback into DMIs has been stressed in numerous previous studies [28, 29]. Vibrotactile feedback has been found to increase controllability of certain musical processes [30]. Haptic feedback will be used as a complement or to reinforce the emitted sounds in the Sound Forest, thereby enhancing the player's musical experience. We suggest to re-produce the situation of a regular acoustic instrument in which the instrument's body amplifies the sound produced by the vibration of the sound-generating mechanism (in our case, the vibration of the augmented string). The idea is to re-recreate such a closed loop between user interaction and haptic rendering by placing a vibrating wooden platform underneath the augmented string. The platform will react to gestures performed on the string. Vibrating floor surfaces are accessible to a wide range of users [31] and therefore go well in line with the design constraints placed upon the Sound Forest.

The haptic floor should be designed to fulfill the following requirements:

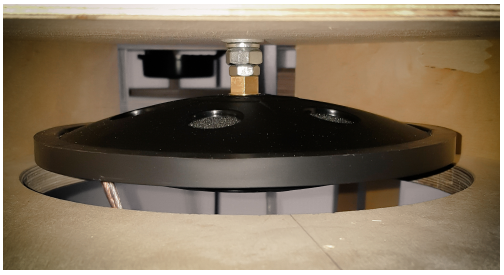
- Provide low-latency feedback on the interaction with the string
- Enable tactile translation [32] of musical sounds emitted by the DMI as well as tactile synthesis of customized haptic feedback
- Transmit frequencies that overlap with the sensitivity domain of FA II receptors in the feet
- Produce perceptually relevant feedback for a wide range of visitors (despite floor deformation due to weight)
- Produce enjoyable vibrotactile feedback: ensure safe whole body vibration, according to the ISO 2631 standard [4]
- Reduce transmission of undesirable structure borne noise: has to be fully decoupled from the floor around and beneath it

A prototype of the first iteration of the vibrating platform can be seen in Figure 3. The prototype consists of a birch-plywood circular surface with a radius of 60 cm under which vibrating actuators (one Clark Synthesis TST239 Silver Tactile Transducer Bass Shaker and one Sinus Live BassPump III bass shaker, for comparative purposes) are fixed on each side of the string. In order to fully understand how to display vibrations to the player in a meaningful way, so that the haptic feedback is tightly coupled to the music, we must first investigate the physical capabilities of

[4] http://www.iso.org/iso/home/store/catalogue_tc/catalogue_detail.htm?csnumber=7612



(a)



(b)

Figure 3: Prototype of the first iteration of the vibrating platform. (a) Vibrating platform, removed from the main floor structure. (b) Clark Synthesis TST239 Silver Tactile Transducer Bass Shaker mounted in the platform structure, seen from the side.

the vibrating platform. As expected, initial measurements done to characterize the frequency response of the platform showed some undesirable resonance peaks that were audible at high amplitudes. Accelerometer measurements on sweeping sinusoids also indicated that the structure of the platform might be too rigid to be fully excited by the current actuator setup. A continuation of the work involves exploring different setups using a tile structure with damping rubber feet, which will both allow the structure to vibrate freely while at the same time support the weight of multiple players standing on top of the platform. The future version of the platform will of course make use of two identical actuators; this will allow for exploration of phase and time delays in order to emphasize specific platform resonances.

4.1.3 Lighting Design

In order to provide a stronger feedback to users while interacting with the strings, we plan to use a LED light intertwined fiber optic cable with DMX controller. Each string, anchored on both floor and ceiling, will change light according to the physical excitation that it receives (e.g. plucking, scraping), with real-time response to user actions. The string should be designed with the following requirements in mind: change colour/intensity/frequency of the lighting feedback in real time when touched/moved by the player, reflect the real-time changing sound properties with changes in the lighting scenario and be robust to different interaction strategies by the museum visitors (e.g. climbing, hanging, strong percussive and plucking

gestures). When the first prototype of the string has been implemented and the overall lighting concept has been accepted, focus will shift to the detailed lighting design solution for the whole room. The room lighting design will follow the following requirements:

- Give an overall perception of the soundscape
- Provide an "idle" status of the room
- Engage/raise interaction of the audience
- Allure people to access the room

4.2 Evaluation

During the first six months of prototyping, an initial string prototype was constructed and evaluated through experiments with a set of users who were allowed to spontaneously explore the string. The design process was iterative: starting from the initial idea, a number of low-fidelity prototypes were developed, formally evaluated, and refined using the collected feedback. Results from these experiments are reported in [33] in which we analyse the types of interaction that were found for users of different age groups (from children to adults) by applying conventional HCI methods.

5. FUTURE DEVELOPMENTS

Once the final installation will be deployed, we will conduct a field investigation from the point of view of the visitor experience. We will apply a multi-method evaluation strategy [34] of different techniques to examine the audience behavior (e.g. log-data analysis, video-cued recall, interviews, questionnaires, observation studies). These formal evaluations techniques will be adopted in order to investigate how appreciated the installation is by visitors at the museum as well as to evidence potential strengths and weaknesses of the system. Findings will be used to adapt the system and to contribute to new knowledge on visitor experiences with interactive artworks, the latter being something we consider to be important given the increasing interest of the interaction design community in the field of interactive art. The Performing Arts Museum and the Sound Forest installation will be inaugurated in early 2017. About 12 000 visitors per month are expected to visit the museum.

Acknowledgments

This project is partially funded by a grant to Roberto Bresin by KTH Royal Institute of Technology, and by Musikverket - Scenkonstmuseet/Swedish Museum of Performing Arts.

6. REFERENCES

[1] S. S. Bautista, *Museums in the Digital Age*. AltaMira Press,U.S., 2013.

[2] J. H. Falk and L. D. Dierking, *The Museum Experience Revisited*. Left Coast Press, 2013.

[3] E. R. Miranda and M. M. Wanderley, *New digital musical instruments: control and interaction beyond the keyboard*. AR Editions, Inc., 2006, vol. 21.

[4] G. D'Arcangelo, "Creating contexts of creativity: musical composition with modular components," in *Proceedings of the 2001 Conference on New Interfaces for Musical Expression, NIME 2001*. National University of Singapore, 2001, pp. 1–4.

[5] T. Blaine and S. Fels, "Contexts of collaborative musical experiences," in *Proceedings of the 2003 Conference on New Interfaces for Musical Expression, NIME 2003*. National University of Singapore, 2003, pp. 129–134.

[6] P. Cook, "Principles for designing computer music controllers," in *Proceedings of the 2001 Conference on New Interfaces for Musical Expression, NIME 2001*. National University of Singapore, 2001, pp. 1–4.

[7] D. Wessel and M. Wright, "Problems and prospects for intimate musical control of computers," *Computer Music Journal*, vol. 26, no. 3, pp. 11–22, 2002.

[8] D. Gentner and J. Nielsen, "The anti-mac interface," *Communications of the ACM*, vol. 39, no. 8, pp. 70–82, 1996.

[9] M. Csikszentmihalyi, "Toward a psychology of optimal experience," in *Flow and the foundations of positive psychology*. Springer, 2014, pp. 209–226.

[10] J. McDermott, T. Gifford, A. Bouwer, and M. Wagy, "Should music interaction be easy?" in *Music and Human-Computer Interaction*. Springer, 2013, pp. 29–47.

[11] P. Oliveros, L. Miller, J. Heyen, G. Siddall, and S. Hazard, "A musical improvisation interface for people with severe physical disabilities," *Music and Medicine*, vol. 3, no. 3, pp. 172–181, 2011.

[12] V. Matossian and R. Gehlhaar, "Human instruments: Accessible musical instruments for people with varied physical ability." *Annual review of Cybertherapy and Telemedicine*, vol. 13, pp. 200–205, 2015.

[13] K. Samuels, "The Meanings in Making: Openness, Technology and Inclusive Music Practices for People with Disabilities," *Leonardo Music Journal*, vol. 25, pp. 25–29, 2015.

[14] Samuels, Koichi, "Enabling creativity: Inclusive music interfaces and practices," in *Proceedings of International Conference on Live Interfaces (ICLI)*, 2014.

[15] S. Katan, M. Grierson, and R. Fiebrink, "Using Interactive Machine Learning to Support Interface Development Through Workshops with Disabled People," in *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*, ser. CHI '15. New York, NY, USA: ACM, 2015, pp. 251–254. [Online]. Available: http://doi.acm.org/10.1145/2702123.2702474

[16] P. M. Rodrigues, R. Gehlhaar, L. M. Girao, and R. Penha, "instruments for everyone: empowering disabled creators with tools for musical expression," *Ubiquity: The Journal of Pervasive Media*, vol. 1, no. 2, pp. 171–191, 2012.

[17] J. McElligott and L. Van Leeuwen, "Designing sound tools and toys for blind and visually impaired children," in *Proceedings of the 2004 conference on Interaction design and children: building a community*. ACM, 2004, pp. 65–72.

[18] K. McCord, "Children with special needs compose using music technology," *Journal of Technology in Music Learning*, vol. 1, no. 2, pp. 3–14, 2002.

[19] A. Barenca and G. Torre, "The manipuller: Strings manipulation and multi-dimensional force sensing." in *Proceedings of the 2011 Conference on New Interfaces for Musical Expression, NIME 2011*, 2011, pp. 232–235.

[20] A. Barenca and M. Corak, "The manipuller ii: Strings within a force sensing ring." in *Proceedings of the 2014 Conference on New Interfaces for Musical Expression, NIME 2014*, 2014, pp. 589–592.

[21] V. Krefeld and M. Waisvisz, "The hand in the web: An interview with Michel Waisvisz," *Computer Music Journal*, vol. 14, no. 2, pp. 28–33, 1990.

[22] M. A. Baalman, "The strimidilator: a string controlled midi-instrument," in *Proceedings of the 2003 Conference on New Interfaces for Musical Expression, NIME 2003*. National University of Singapore, 2003, pp. 19–23.

[23] C. Unander-Scharin, Å. Unander-Scharin, K. Höök, and L. Elblaus, "Interacting with the vocal chorder: re-empowering the opera diva," in *CHI'14 Extended Abstracts on Human Factors in Computing Systems*. ACM, 2014, pp. 603–606.

[24] A. Tanaka, "Musical performance practice on sensor-based instruments," *Trends in Gestural Control of Music*, vol. 13, no. 389-405, p. 284, 2000.

[25] R. I. Godøy and M. Leman, *Musical Gestures: Sound, Movement, and Meaning*. Routledge, 2009.

[26] S. Dahl, F. Bevilacqua, R. Bresin, M. Clayton, L. Leante, I. Poggi, and N. Rasamimanana, "Gestures in performance," in *Musical Gestures: Sound, Movement, and Meaning*. Routledge, 2009, pp. 36–68.

[27] G. Weinberg, "Interconnected musical networks: Toward a theoretical framework," *Computer Music Journal*, vol. 29, no. 2, pp. 23–39, 2005.

[28] D. M. Birnbaum and M. M. Wanderley, "A systematic approach to musical vibrotactile feedback," in *Proceedings of the International Computer Music Conference (ICMC)*, vol. 2, 2007, pp. 397–404.

[29] R. Pedrosa and K. MacLean, "Perceptually informed roles for haptic feedback in expressive music controllers," in *Haptic and Audio Interaction Design*. Springer, 2008, pp. 21–29.

[30] M. T. Marshall and M. M. Wanderley, "Vibrotactile feedback in digital musical instruments," in *Proceedings of the 2006 Conference on New Interfaces for Musical Expression, NIME 2006*. IRCAM–Centre Pompidou, 2006, pp. 226–229.

[31] Y. Visell, A. Law, and J. R. Cooperstock, "Touch is everywhere: Floor surfaces as ambient haptic interfaces," *Haptics, IEEE Transactions on*, vol. 2, no. 3, pp. 148–159, 2009.

[32] M. Giordano and M. M. Wanderley, "Perceptual and technological issues in the design of vibrotactile-augmented interfaces for music technology and media," in *International Workshop on Haptic and Audio Interaction Design*. Springer, 2013, pp. 89–98.

[33] J. Paloranta, A. Lundström, L. Elblaus, R. Bresin, and E. Frid, "Interaction with a large sized augmented string instrument intended for a public setting," in *Proceedings of SMC 2016 - 13th International Conference on Sound and Music Computing, Hamburg, 31 August & September 1-3, 2016*, 2016.

[34] E. Hornecker and M. Stifter, "Learning from interactive museum installations about interaction design for public settings," in *Proceedings of the 18th Australia conference on Computer-Human Interaction: Design: Activities, Artefacts and Environments*. ACM, 2006, pp. 135–142.

GESTURECHORDS: TRANSPARENCY IN GESTURALLY CONTROLLED DIGITAL MUSICAL INSTRUMENTS THROUGH ICONICITY AND CONCEPTUAL METAPHOR

Dom Brown, Chris Nash, Tom Mitchell
Department of Computer Science and Creative Technologies
University of the West of England
Bristol, UK
[dom.brown, chris.nash, tom.mitchell]@uwe.ac.uk

ABSTRACT

This paper presents GestureChords, a mapping strategy for chord selection in freehand gestural instruments. The strategy maps chord variations to a series of hand postures using the concepts of iconicity and conceptual metaphor, influenced by their use in American Sign Language (ASL), to encode meaning in gestural signs. The mapping uses the conceptual metaphors MUSICAL NOTES ARE POINTS IN SPACE and INTERVALS BETWEEN NOTES ARE SPACES BETWEEN POINTS, which are mapped respectively to the number of extended fingers in a performer's hand and the abduction or adduction between them. The strategy is incorporated into a digital musical instrument and tested in a preliminary study for transparency by both performers and spectators, which gave promising results for the technique.

1. INTRODUCTION

When designing Digital Musical Instruments (DMIs), the mapping strategy used to connect a performer's actions to an auditory response is of critical importance, and can "define the very essence of an instrument" [1]. As such, a great amount of research has gone into designing mapping strategies that provide and enable an expressive [2, 3] and virtuosic performance [4].

This paper seeks to explore the use of conceptual metaphors in freehand DMI mapping by their iconic representation, drawing influence from gestural sign languages such as American Sign Language (ASL). An iconic representation is the use of resemblance or similarity to encode meaning [5], and many gestures in ASL use it to encode conceptual metaphors [6]. Some signs that use this method of encoding their meaning can often be comprehended by those with no experience in signed languages [7] due to their physical resemblance to the concept they represent. This paper seeks to begin investigating the prospects of using similar techniques to encode musical meaning in freehand gestural control, specifically focussing on whether using this technique in DMI mapping design provides a high level of "transparency" as defined by Fels *et al.*, which "provides an indication of the psychophysiological distance, in the minds of the player and the audience, between the input and output of a device mapping" [8]. More succinctly: Does using iconic representation of conceptual metaphor in mapping strategies make for effective, transparent control in freehand gestural musical instruments?

To explore this, a DMI mapping strategy, GestureChords, has been created. This strategy encodes conceptual metaphors relating to musical concepts in a series of hand postures, via iconic representation, to be used in DMIs for selecting chord variations. This strategy is then incorporated into a DMI, and tested for its transparency.

2. BACKGROUND

Due to the uncoupling of a musician's actions and the resulting audio response, DMIs need a specified mapping strategy in order to re-establish the connection, which can come to define a musical instrument [1]. Thus, developing a successful mapping strategy is of the utmost importance in DMI design.

2.1 Mapping

The strategies used to map input data to musical parameters in DMIs commonly fall within one-to-one, divergent (one-to-many), convergent (many-to-one) and many-to-many classifications [9, 10]. However, the most successful and expressive mapping strategies have been found to be those that employ multi-parametric control with a high degree of complexity [11].

This desire for complexity, as well as advances in gestural recognition technologies [12–14] has lead to the emergence of more abstract applications of mapping, as highly complex strategies can be devised independently and then taught to computers using machine learning techniques [15]. While this has lead to the ability to make complex mappings with relative ease, there still remain many challenges to be overcome. Notably, "how are meaningful and effective mappings created, that seem to evoke the *correct* musical response?"

One solution to this issue is to allow a musician to decide on their own mappings [16, 17]. While this provides

a meaningful mapping for the individual performer who designed them, this does not necessarily mean that another musician would find these mappings intuitive, nor an audience member in any performances given, whose perception of a performance plays an important role in instrument design [18]. This technique also requires a lengthy setup process on the part of the performer, and mappings may also need to be set in a prescribed order, requiring premature commitment from the performer [19] as they may be difficult to alter later.

2.2 Conceptual Metaphor

The use of the term metaphor in Human-Computer Interaction (HCI) has a large scope of potential meanings and uses, and requires contextualisation [20]. Here, the term refers to conceptual metaphor, or when one concept is explained in the terms of another [21]. It is a useful concept in HCI for explaining the behaviour of computer software, and allows users to grasp abstract concepts quickly via an association with a more familiar domain [22]. Using it provides a way to "piggyback" understanding of abstract concepts on the structure of concrete concepts [23]. A classic example of this is the DELETING IS RECYCLING conceptual metaphor, in which files users wish to delete are temporarily stored in a specific directory named "Recycle Bin" (on Windows operating systems), which then "recycles" the material it is made from (in the computer's case, memory instead of paper).

This application of metaphor has also been explored in DMI design [4, 8, 24]. Fels *et al.* [8] and Wessel and Wright [4] examine the effectiveness of using conceptual metaphors in instrument mapping design to allow for expressive and virtusoic performance. Wessel and Wright use conceptual metaphors described by Lakoff and Johnson [21] to influence the design of several instruments, while Fels *et al.* describe how this can be used to increase the "transparency" of the instrument's mapping. Here, transparency describes how comprehensible the mapping is to a player and observer (ranging from "opaque" to "transparent"), a quality that contributes to an instrument's expressive and virtuosic potential. A similar concept is explored by Reeves *et al.* in more general HCI contexts, particularly focussing on a spectator's ability to perceive a user's "manipulations" and the resulting "effects", on a scale from "hidden" to "amplified" [25]. It is important to consider the spectator's understanding of an instrument's mapping as well as the performer's in DMI design, as the ability for an audience to perceive how an instrument is controlled is a critical aspect of musical performance [18, 26].

HCI and DMI design are not the only domains to make use of conceptual metaphor. In fact, conceptual metaphor is a tool so ubiquitous that it is used reflexively (without conscious thought) [21], and is common in natural language. For example, the conceptual metaphor ARGUMENT IS WAR described by Lakoff and Johnson: "Your claims are *indefensible*", "He *attacked every weak point* in my argument" and so on.

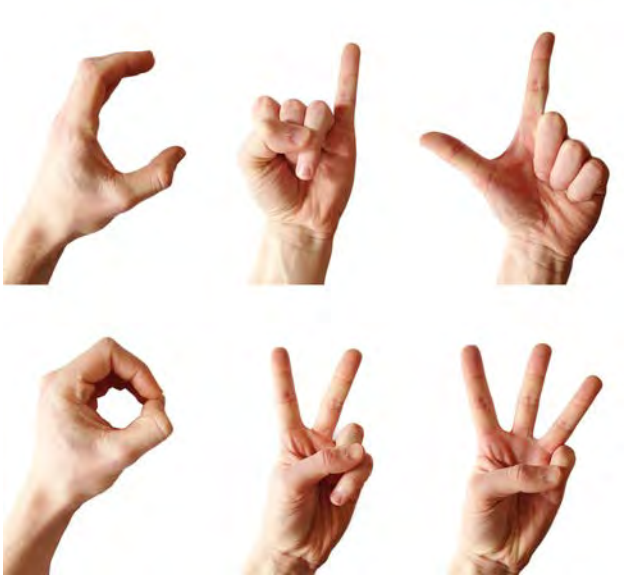


Figure 1: ASL fingerspelling letters: 'C', 'I', 'L', 'O', 'V' and 'W'.

In the design of DMIs that use freehand gestures as their interaction method, the most useful derivation of linguistic conceptual metaphor is through its prevalence in freehand gestural languages, or sign languages.

2.3 Iconicity and Conceptual Metaphor in Sign Language

Iconicity is found in signs that represent their objects mainly by their similarity, or perceived resemblance, no matter what their mode of being [5]. The use of iconicity to encode meaning is common in gestural sign languages. In this case, signs visually resemble that which they represent, enabling them, in some cases, to be recognised by non-signers [7].

Examples of this can be found in the ASL fingerspelling alphabet. This alphabet is a system of 24 static hand postures and two dynamic gestures used to encode the standard English alphabet, all performed on one hand. These postures can be said to be *emblematic*, which refers to nonverbal acts which have a direct verbal translation, for which a precise meaning is known by most or all members of a group or culture [27]. Emblematic postures and gestures are often iconically encoded, and many of the ASL letters are iconic representations of their written counterparts; the hand shapes used to encode them physically resemble the shapes of the letters, such as 'C', 'I', 'L', 'O', 'V' and 'W' (Figure 1). The postures can be signed on either hand, and are expressed on the left hand as a mirror image of the right.

Conceptual metaphors are also regularly expressed through iconic representation in sign languages. An example, described by Taub [6], is the conceptual metaphors of INFORMATION ARE OBJECTS and HEAD IS A CONTAINER, which are iconically expressed in the ASL sign LEARN (Figure 2), in which the signer gestures the picking up of information and the placing of it in

Figure 2: The ASL sign LEARN

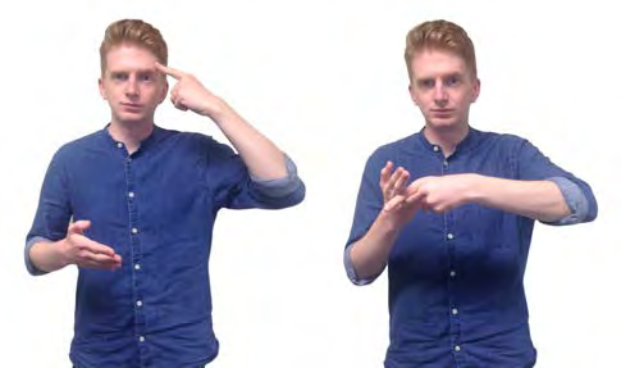


Figure 3: The ASL sign THINK-PENETRATE

one's head. Another is the ASL sign THINK–PENETRATE (Figure 3). This sign begins with the dominant hand pointing with the index finger at the temple, which then moves through or *penetrates* the fingers of the non-dominant hand. This sign can be interpreted as "they finally got the point" and makes use of the same metaphors as the sign LEARN, elaborating on the HEAD IS A CONTAINER metaphor with CONTAINERS HAVE BOUNDARIES, while INFORMATION ARE OBJECTS leads to INFORMING IS SENDING. The sign iconically depicts the information object (the thought) being sent from one container (the signer's head) to the boundary of another container (the signer's hand, representing another's head), penetrating it and entering (the thought enters the head).

3. GESTURECHORDS

The mapping in GestureChords is based on an iconic representation of conceptual metaphors. Particularly, the metaphors of MUSICAL NOTES ARE POINTS IN SPACE and INTERVALS BETWEEN NOTES ARE SPACES BETWEEN POINTS. These metaphors have been inferred as follows: Music is experienced through time; time is expressed through spatial metaphors (TIME IS A MOVING OBJECT [21]); thus, notes are points in this musical space that are reached as we travel through it (or it travels past us). As notes are experienced, they are identified via differences in pitch; difference in pitch is often expressed in spatial terms (such as UP–DOWN [24]); thus, differences in pitch between notes (or intervals) are distances between points.

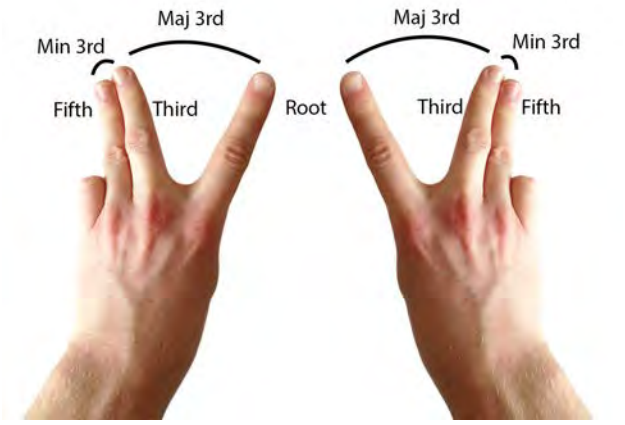


Figure 4: A major triad chord represented using the GestureChords strategy, as expressed on both the left and right hands.

These metaphors can also be said to be expressed in Western musical notation: one travels through the music from left to right; notes are represented by black points on the stave, whose position on the up–down axis on ledger lines denotes pitch; while the intervals between notes are represented by the distance (on the vertical axis) between these points.

The GestureChords system of hand postures uses the conceptual metaphors above to encode chord shapes that are intended for use in free hand DMIs. The mapping strategy considers the number of extended fingers on the hand and the spacing (abduction) between them. The MUSICAL NOTES ARE POINTS IN SPACE metaphor is iconically mapped to the tips (or points) of extended fingers, while INTERVALS BETWEEN NOTES ARE SPACES BETWEEN POINTS is iconically represented by the spaces between the fingers. As such, each extended finger represents one note in the resulting chord, while the adduction or abduction between consecutive fingers represents one of two intervals between these notes. Adducted (close together) fingers represent minor thirds (the small gap representing the smaller interval encoded) while abducted (spread out) fingers represent major thirds (the large gap representing the larger interval). As in ASL fingerspelling, the mapping is designed to be used with both hands, with signs expressed on one hand as a mirror image of their expression on the other. Accordingly, the index finger always represents the root note, while subsequent fingers represent notes above it. The thumb is excluded from the mapping.

For example, a major triad chord is represented by the hand posture in Figure 4. The three extended fingers represent the three notes used (e.g. C–E–G in C Major). The abducted index and middle fingers encode the major third between the root and the third of the chord (C–E), while the adducted middle and ring fingers encode the minor third between the third and the fifth of the chord (E–G).

The encoding above constrains GestureChords to representing a maximum of four note chords, and encodes 14 different chord types.

The full range of hand postures is shown in Figure 5. The choice of these postures are the natural result of following the strategy set out above. It should be noted that the final possible hand posture of four abducted fingers has been omitted, as in this mapping it represents an augmented chord, which is already represented, in an alternative voicing. The mapping encodes root, minor third, major third, diminished, minor, major, augmented, diminished seventh, diminished major seventh, minor seventh, minor major seventh, dominant seventh, major seventh and augmented major seventh chords.

Figure 5: The full range of hand postures used to select chords in GestureChords.

4. PILOT STUDY

To evaluate the efficacy of the GestureChords mapping strategy a DMI was built that incorporates a Leap Motion optical sensor [28] and a simple one octave virtual keyboard embedded in the instrument's software user interface (Figure 6). A user interacts with the instrument by positioning one hand above the Leap Motion sensor to use the chord postures, while using a mouse with their other hand to interact with the virtual keyboard (Figure 7). The software analyses the Leap Motion's input using an Adaptive Naïve Bayes Classification algorithm from The Gesture Recognition Toolkit [29] to determine which chord has been selected. The virtual keyboard then selects the root note and triggers the chord. The application provides visual feedback, informing the user as to which chord and root note is currently selected, as well as the connection status of the Leap Motion.

Note selection is a difficult challenge for freehand gestural instruments. Previous studies [30, 31] have shown that this is often due to a lack of tactile and visual feedback, usually given by a physical interaction surface found on traditional instruments. The decision to select root notes and trigger the chords on a virtual keyboard has been made in order to avoid these issues and focus the attention of the study on the GestureChords postures.

4.1 Methodology

In this pilot study, responses from participants in a qualitative study are compared against the transparency scale described by Fels *et al.* (Figure 8) [8] to give an indication of the transparency of the GestureChords mapping. The scale consists of two axes ranging from opaque to transparent, one for the performer's perception and the other for their audience. Successful mappings are those that score highly on both axes, transparent for both the performer and their audience. The study is split into two tests, one for each axes and each with its own set of participants. In both tests, the musical expertise of the participant is established by asking for an explanation of the theory behind major, minor, augmented, diminished, minor seventh and major seventh chords.

The technique used in the performer test draws from the discourse analysis technique described by Stowell *et al.* [32], which consists of: *free exploration*, where a user is allowed to explore the instrument freely; *guided exploration*, where a user is asked to influence their exploration from an example performance; and a *semi-structured interview*, where the user's subjective experience is evaluated. The method implemented in this pilot study consisted of free exploration, guided exploration and a questionnaire. The questionnaire is used to focus the participants responses to the mapping strategy, and gauge its transparency with regards to the performer's perceptions.

The methodology for the audience test is adapted from the spectator evaluation technique described by Barbosa *et al.* [33]. In this technique, a video of a performance is presented to participants along with a questionnaire for analysing the participant's comprehension of *cause, effect, mapping, intention* and *error*. In this test, participants are

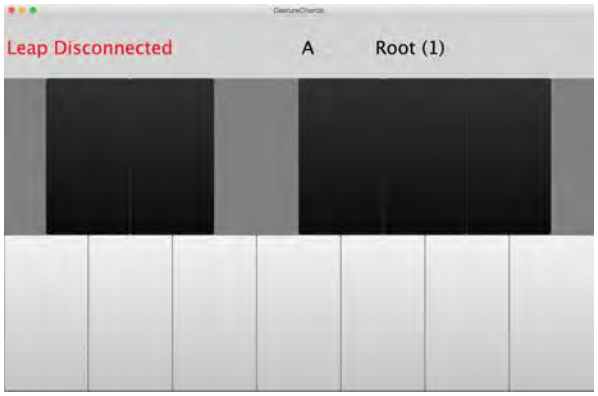


Figure 6: The GestureChords application.

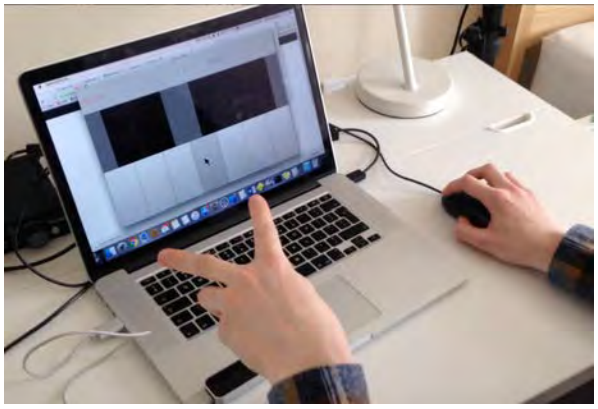


Figure 7: A GestureChords performance.

shown a video of a performance with the GestureChords application and asked a series of questions, which in this study focus on the comprehension of *cause, effect* and *mapping*, in order to determine an audience's perception of the mapping's transparency.

The video allowed the participants to clearly see the GestureChords hand postures being performed as well as the performer's interactions with the software interface.

In both tests, a full description of the mapping strategy was initially withheld, and then revealed to the participant midway through the test. This was done to compare the participant's perception of the mapping with and without knowledge of the strategy employed, and to test if they were able to independently perceive the iconic representation of the conceptual metaphors without prompting.

4.2 Results

Six participants took part in the performer test, while four took part in the audience test. In both tests the participants ranged from musicians with advanced knowledge of chord theory to relative novices, whose descriptions of major and minor chords did not extend further than informal observations, such as "major is happy" and "minor is sad".

4.2.1 Performer

All of the performance participants agreed that the method for controlling the instrument was clear, and all recognised

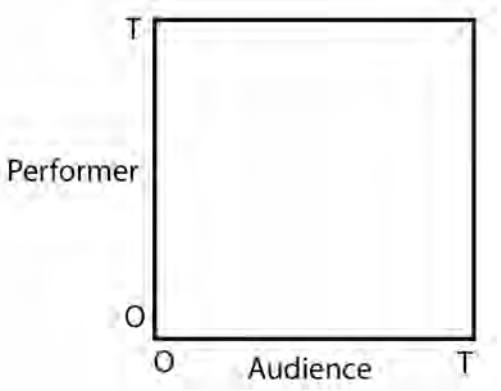


Figure 8: The Mapping Transparency Scale [8].

that different hand postures triggered different chords, while the virtual keyboard selected the root note and triggered the chord.

Three users, two of which displayed advanced knowledge of musical theory while the other had limited knowledge, managed to recognise and describe the mapping strategy before it was revealed to them, noting that abduction and adduction variation mapped to major and minor thirds, while each finger added a note to the chord. Two participants simply noted that different postures triggered different chords, while one participant offered no description.

Participants noted that the instrument was "intuitive" and "entertaining to play", and two users remarked that they believed the instrument would be "suitable for beginners in music theory".

Once the mapping strategy was revealed, all the participants strongly agreed that they understood the concept, while five of the six agreed that the iconic representation aided in their understanding of the mapping. All the participants agreed that the mapping was an effective method of representing chords.

4.2.2 Audience

Corresponding to *cause* and *effect* comprehension, all of the audience test participants agreed that the method of controlling the instrument was clear. Two respondents were able to give detailed responses on how they perceived the instrument to be controlled and how the resulting auditory response was achieved, while two users was unable to fully perceive the controls, and gave vague responses.

Regarding *mapping* comprehension, three of the four participants agreed that the controls clearly related to the auditory response. All the participants correctly recognised the mapping of the number of notes in the chord to the number of extended fingers, while two participants managed to recognise the relation between abduction/adduction and major and minor third intervals.

Once the mapping strategy had been revealed, all the participants agreed that the relationship between a chord and its hand posture was easily perceived, and that the strategy was effective at encoding chords.

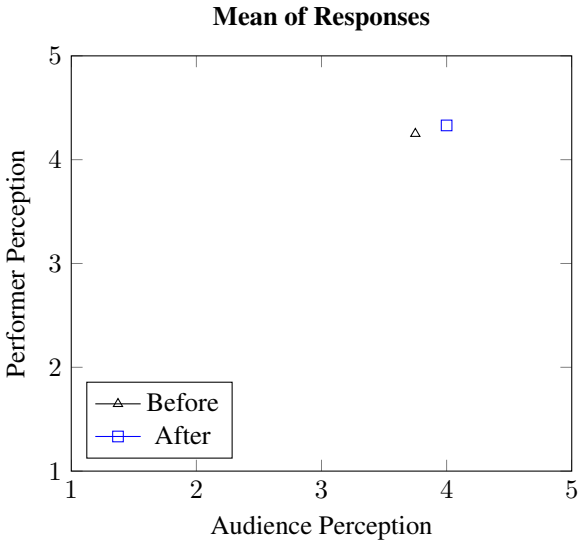


Figure 9: Mapping Transparency of GestureChords

4.3 Discussion

The results of this initial pilot study show positive results for GestureChords and the use of iconic representation of conceptual metaphor in DMI mapping design.

It should be noted that due to the small size of this study these results cannot be considered conclusive, a more detailed and thorough investigation into the technique is required in order to determine its true effectiveness. However, the positive results from this study are promising, and suggest that using iconic representation of conceptual metaphor in gestural musical instrument mapping design can promote a transparent mapping strategy for both performers and their audiences, and that further exploration into this technique is worthwhile.

An interesting observation that arose was that the system may be appropriate for music theory novices. This highlights an area of possible future research, and may relate to users being able to use GestureChords to cognitively offload [34] the concepts of chord selection to their hands, allowing the postures to become epistemic actions [35].

Both performers and audiences were asked to rate their opinion from 1 (opaque) to 5 (transparent) on how obvious they perceived the mapping to be. The mean averages of these responses have been mapped onto the Mapping Transparency Scale of Fels *et al.* [8] for both before and after the mapping was revealed to participants, shown in Figure 9. Figure 10 shows the spread of responses in percentages. This preliminarily rating can provide a rough guide, and suggests that the GestureChords mapping strategy has been successful in providing a transparent mapping strategy for freehand gestural control of chords, and that the mapping was perceived to be transparent prior to participants gaining knowledge of the mapping strategy as well as after. This suggests that prior knowledge about how an instrument is played may be irrelevant when the mapping strategy uses iconic representation. However, a more thorough study is called for to explore

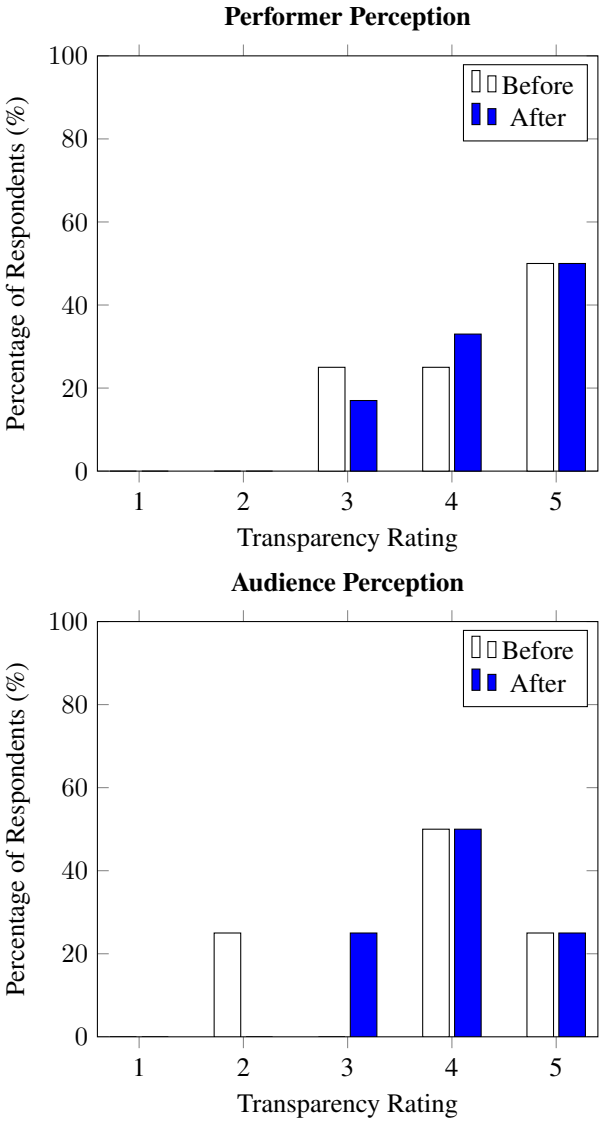


Figure 10: Transparency Ratings of GestureChords

this hypothesis further. A larger sample size would allow for a useful application of more detailed statistical analysis, such as calculating the variance and standard deviation of perceptions. It would also allow for further exploration into the influence of prior musical knowledge on a participant's ability to infer the mapping strategy.

5. CONCLUSIONS AND FUTURE WORK

The GestureChords mapping strategy has been presented, which uses iconic representations of musical conceptual metaphors to provide a transparent mapping of chord selection for freehand gestural control. The strategy has been incorporated in a simple DMI and given a preliminary analysis to test for the mapping's transparency. This pilot study suggests that the mapping successfully provides a transparent mapping for both audiences and performers, and shows promising results for the use of iconic representations of conceptual metaphors in the mapping design of freehand digital musical instruments.

Further developments from this paper will include the development of more complex gestural musical instrument mapping strategies using the iconic representation of conceptual metaphor technique. This will include: exploring note excitation as well as modification, moving away from the reliance on existing instrument metaphors and into pure freehand mapping; applying the technique to dynamic gestural control using continuous movement, allowing for musical expression to be realised in finer detail; and performing further evaluations.

Acknowledgments

The authors would like to thank the participants who gave their time to take part in this study.

6. REFERENCES

[1] A. Hunt, M. M. Wanderley, and M. Paradis, "The importance of parameter mapping in electronic instrument design," *Journal of New Music Research*, vol. 32, no. 4, pp. 429–440, 2003.

[2] D. Arfib, J.-M. Couturier, and L. Kessous, "Expressiveness and digital musical instrument design," *Journal of New Music Research*, vol. 34, no. 1, pp. 125–136, 2005.

[3] C. Dobrian and D. Koppelman, "The 'e' in nime: musical expression with new computer interfaces," in *Proceedings of New Interfaces for Musical Expression (NIME 2006)*, Paris, France, 4th–8th June 2006, pp. 277–282.

[4] D. Wessel and M. Wright, "Problems and prospects for intimate musical control of computers," *Computer Music Journal*, vol. 26, no. 3, pp. 11–22, 2002.

[5] C. S. Peirce, *The Philosophy of Peirce: Selected Writings*, J. Buchler, Ed. Dover Publications Inc., 1986.

[6] S. F. Taub, *Language from the body: Iconicity and metaphor in American Sign Language*. Cambridge University Press, 2001.

[7] A. K. Lieberth and M. E. B. Gamble, "The role of iconicity in sign language learning by hearing adults," *Journal of Communication Disorders*, vol. 24, no. 2, pp. 89–99, 1991.

[8] S. Fels, A. Gadd, and A. Mulder, "Mapping transparency through metaphor: towards more expressive musical instruments," *Organised Sound*, vol. 7, no. 2, pp. 109–126, 2002.

[9] J. B. Rovan, M. M. Wanderley, S. Dubnov, and P. Depalle, "Instrumental gestural mapping strategies as expressivity determinants in computer music performance," in *Proceedings of Kansei - The Technology of Emotion Workshop*, Genova, Italy, 3rd–4th October 1997, pp. 3–4.

[10] A. Hunt and M. M. Wanderley, "Mapping performer parameters to synthesis engines," *Organised Sound*, vol. 7, no. 2, pp. 97–108, 2002.

[11] A. D. Hunt, "Radical user-interfaces for real-time musical control." Ph.D. dissertation, University of York, 1999.

[12] S. Mitra and T. Acharya, "Gesture recognition: A survey," *IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews*, vol. 37, no. 3, pp. 311–324, 2007.

[13] F. Bevilacqua, B. Zamborlin, A. Sypniewski, N. Schnell, F. Guédy, and N. Rasamimanana, "Continuous realtime gesture following and recognition," in *Gesture in Embodied Communication and Human-Computer Interaction*. Springer, 2010, pp. 73–84.

[14] B. Caramiaux, N. Montecchio, A. Tanaka, and F. Bevilacqua, "Adaptive gesture recognition with variation estimation for interactive systems," *ACM Transactions on Interactive Intelligent Systems (TiiS)*, vol. 4, no. 4, p. 18, 2014.

[15] B. Caramiaux and A. Tanaka, "Machine learning of musical gestures." in *Proceedings of New Interfaces for Musical Expression (NIME 2013)*, Seoul, South Korea, 27th–30th May 2013, pp. 513–518.

[16] J. Françoise, "Gesture–sound mapping by demonstration in interactive music systems," in *Proceedings of the 21st ACM international conference on Multimedia*. Barcelona, Spain: ACM, 21st–25th October 2013, pp. 1051–1054.

[17] B. Caramiaux, J. Françoise, N. Schnell, and F. Bevilacqua, "Mapping through listening," *Computer Music Journal*, vol. 38, no. 3, pp. 34–48, 2014.

[18] A. C. Fyans and M. Gurevich, "Perceptions of skill in performances with acoustic and electronic instruments." in *Proceedings of New Interfaces for Musical Expression (NIME 2011)*, Oslo, Norway, 30th May–1st June 2011, pp. 495–498.

[19] C. Nash, "The cognitive dimensions of music notations," in *Proceedings of the First International Conference on Technologies for Music Notation and Representation (TENOR 2015)*. Paris, France: Institut de Recherche en Musicologie, May 2015, pp. 190–202.

[20] A. F. Blackwell, "The reification of metaphor as a design tool," *ACM Transactions on Computer-Human Interaction (TOCHI)*, vol. 13, no. 4, pp. 490–530, 2006.

[21] G. Lakoff and M. Johnson, *Metaphors We Live By*. University of Chicago Press, 1980.

[22] A. Marx, "Using metaphor effectively in user interface design," in *Conference Companion on Human Factors in Computing Systems*. Boston, MA, USA: ACM, 24th–28th April 1994, pp. 379–380.

[23] M. S. McGlone, "What is the explanatory value of a conceptual metaphor?" *Language & Communication*, vol. 27, no. 2, pp. 109–126, 2007.

[24] P. H.-K. Wong, "Conceptual metaphor in the practice of computer music," Master's thesis, Mills College, 2011.

[25] S. Reeves, S. Benford, C. O'Malley, and M. Fraser, "Designing the spectator experience," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. Montreal, Canada: ACM, 22nd–27th April 2005, pp. 741–750.

[26] S. O'Modhrain, "A framework for the evaluation of digital musical instruments," *Computer Music Journal*, vol. 35, no. 1, pp. 28–42, 2011.

[27] P. Ekman and W. V. Friesen, "Hand movements," *Journal of Communication*, vol. 22, no. 4, pp. 353–374, 1972.

[28] "Leap motion," leapmotion.com, 2016, accessed: 4th July 2016.

[29] N. Gillian and J. A. Paradiso, "The gesture recognition toolkit," *The Journal of Machine Learning Research*, vol. 15, no. 1, pp. 3483–3487, 2014.

[30] J. Han and N. Gold, "Lessons learned in exploring the leap motion ™ sensor for gesture-based instrument design," in *Proceedings of New Interfaces for Musical Expression (NIME 2014)*, London, UK, 30th June–4th July 2014, pp. 371–374.

[31] M. Ritter and A. Aska, "Leap motion as expressive gestural interface," 2014.

[32] D. Stowell, M. D. Plumbley, and N. Bryan-Kinns, "Discourse analysis evaluation method for expressive musical interfaces," in *Proceedings of New Interfaces for Musical Expression (NIME 2008)*, Genova, Italy, 5th–7th June 2008, pp. 81–86.

[33] J. Barbosa, F. Calegario, V. Teichrieb, G. Ramalho, and P. McGlynn, "Considering audience's view towards an evaluation methodology for digital musical instruments." in *Proceedings of New Interfaces for Musical Expression (NIME 2012)*, Ann Arbor, MI, USA, 21st–23rd May 2012.

[34] I. E. Dror and S. Harnad, "Offloading cognition onto cognitive technology," *Cognition Distributed: How cognitive technology extends our minds*, vol. 16, no. 1, 2008.

[35] D. Kirsh and P. Maglio, "On distinguishing epistemic from pragmatic action," *Cognitive Science*, vol. 18, no. 4, pp. 513–549, 1994.

AN ONLINE TEMPO TRACKER FOR AUTOMATIC ACCOMPANIMENT BASED ON AUDIO-TO-AUDIO ALIGNMENT AND BEAT TRACKING

Grigore Burloiu
Faculty of Electronics, Telecommunications and Information Technology
University Politehnica of Bucharest
gburloiu@gmail.com

ABSTRACT

We approach a specific scenario in real-time performance following for automatic accompaniment, where a relative tempo value is derived from the deviation between a live target performance and a stored reference, to drive the playback speed of an accompaniment track. We introduce a system which combines an online alignment process with a beat tracker. The former aligns the target performance to the reference without resorting to any symbolic information. The latter utilises the beat positions detected in the accompaniment, reference and target tracks to (1) improve the robustness of the alignment-based tempo model and (2) take over the tempo computation in segments when the alignment error is likely high. While other systems exist that handle structural deviations and mistakes in a performance, the portions of time where the aligner is attempting to find the correct hypothesis can produce erratic tempo values. Our proposed system, publicly available as a Max/MSP external object, addresses this problem.

1. INTRODUCTION

The task of online tempo tracking refers to the computation of a realistic tempo curve from an incoming audio stream in real time, relative to a reference. In this paper we focus on the application of tempo tracking in a musical context where a performer plays together with a responsive accompaniment track, with both the system reacting to live tempo fluctuations and acting as a steady backdrop for the musician to anchor herself to when needed. To this end, we examine two paradigms for tempo tracking—online audio-to-audio alignment and beat tracking—and propose a system that uses both, harnessing their respective strengths.

In the literature, audio alignment is generally part of *score following* systems [1–3], which gradually build a best-fit path matching the incoming live audio to a reference. This path is likely to contain intermittently unnatural slopes, which is why *tempo models* are needed to translate the path slope into a realistic relative tempo value [2]. A typical application for a score following task is a concerto simulation, where a solo performance and drives a machine-generated accompaniment, which faithfully responds to the musician's tempo progression. In cases of performance errors or structural differences such as jumps or repeats, online alignment algorithms can temporarily produce erratic local outputs. Tempo models can alleviate the problem by smoothing over minor errors, but the larger issue of systematic error detection and correction remains.

Meanwhile, beat tracking systems compute tempo curves by detecting the dominant metrical pulse in the live input [4, 5]. As such, they are geared towards scenarios with prominent periodic beats, where the live musician might be locked in with a rhythmic backing track. Wrong notes do not affect the tempo tracking performance, and even when the musician deviates from the rhythmic pattern, beat trackers produce a reasonably consistent tempo line. However, even though they might be configured to drive an external sequencer [4], beat tracking systems themselves do not have information about the material being played. Thus minor deviations can cause them to fall on an upbeat, and structural changes such as the omission of a bar of music are impossible to detect.

We introduce a system that tackles the issue of errors and structural deviations in music with a strong rhythmic pulse, by tracking tempo in two ways: a model based on audio-to-audio alignment by default, and a beat tracking-based model which takes over when the alignment path becomes erratic. The resulting machine can drive an accompaniment track based only on audio inputs, without the use of a score or any other symbolic ground truth information. The practical justification is that, for reasons of expediency, lack of access, or incompatibility with the material, a musician might rather record a reference performance of a part instead of plugging in a MIDI or MusicXML symbolic score.

An early version of our proposed system was presented in [3]. The application and its source are available [1] as a Max/MSP [2] external object; to our knowledge, it is the only online audio-to-audio alignment tool for Max to date.

The rest of the paper is organised as follows: section 2 is a brief review of relevant work. In section 3 we describe our alignment framework and in section 4 we introduce the beat tracking component and describe its integration into the system. Section 5 is a case study of the switching mechanism. We conclude with a discussion of the proposed system and future research directions in section 6.

[1] See https://github.com/RVirmoors/RVdtw-.

[2] Max is a state-of-the-art programming environment for realtime multimedia performance; see http://cycling74.com/.

2. RELATED WORK

Several studies have addressed the problems of performance variation and error, and of structural differences in the context of music alignment. In the case of alignment to a symbolic score, the problem of jumps is relatively easily solved by marking points of possible divergence [6–8]. For the audio-to-audio alignment of one performance to another, which is the focus of our paper, methods based on dynamic time warping (DTW) [9] and variations thereof [10] have shown good results in the offline case. For real-time situations, audio-to-audio alignment has been implemented using strategies based on online DTW [1, 2, 11] or particle filtering [12, 13].

Notably, structural differences are specifically addressed in [2, 13], which continuously monitor different positions in the score in parallel, to account for jumps. These methods, while performing well for tasks such as real-time page turning and annotation, are however not optimised for automatic accompaniment. There is no mechanism to ensure a musically useful tempo during time periods of incorrect alignment caused by mistakes, jumps, or improvisation.

Meanwhile, beat trackers have been at the core of "performance following" [14] tasks such as generative drum accompaniment [4] or tonal performance tracking [14, 15]. Such applications indicate that beat trackers are able to drive a performance forward over periods where the live musician strays from the original reference, which is the main insight driving this paper.

3. PERFORMANCE AUDIO ALIGNMENT

In this section we describe our pre-existing accompaniment framework, as expanded from [3]. It consists of an online audio-to-audio aligner and a tempo model.

3.1 Online DTW-based Follower

Our system computes the alignment path based on a variant of online DTW [1]. The basic algorithm aligns a *target* time series $X = x_1 \dots x_m$ to a *reference* $Y = y_1 \dots y_n$, where X is partially unknown: only the first t values are known at a certain point. The goal is for each $t = 1 \dots m$ to find the corresponding index h_t, so that the subsequence $y_1 \dots y_{h_t}$ is best aligned [3] to $x_1 \dots x_t$. The alignment path p is a sequence of (t, h_t) tuples connecting the origin with the current position for the lowest global match cost. For audio alignment tasks, all x_i and y_j are audio feature vectors; in this case we use chromagrams, as computed in [16].

The alignment path p is defined as being monotonous and continuous, with a local slope constraint that prevents the follower from getting stuck in a local minimum or making steep jumps. For each incoming frame x_t, the local match cost matrix is computed for the past $c \times c$ frames and the path advances by computing a new row, a new column, or both, depending on where the current minimum match cost is found on the search window's frontier: a row, a column or the top-right corner, respectively.

[3] In cases where a frame x_t is assigned to several frames in Y, we set h_t to point to the last of these.

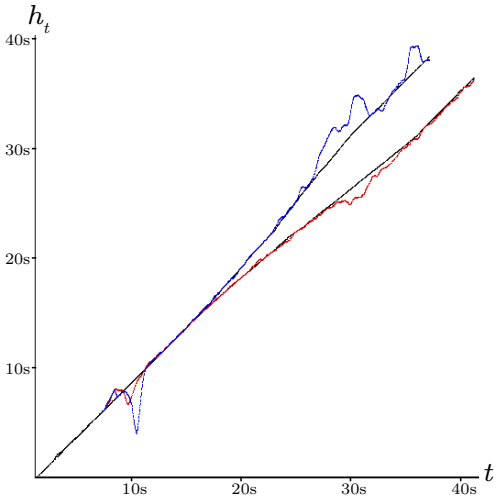


Figure 1: The online DTW-based audio alignment. Target time t progresses horizontally, reference time h_t vertically. Two test target alignment paths are shown in black. The deviations δ_t^{BACK} from the backwards path are added to each (t, h_t) tuple, in blue for the accelerating target and in red for the decelerating one.

Since the alignment engine is not the main focus of this paper, we will just briefly describe our main modifications to the classic online DTW method. For implementation details we refer the reader to [1, 3].

Our first change is to remove the path slope constraint, allowing the alignment path to trace along the X or Y axes for as long as necessary. To maintain stability, we compensate by adjusting the local match weighting coefficients to favour diagonal movement, and by adding a constant α to the local cost, minimising the influence of minor differences between target and reference feature vectors, which could have unpredictably diverted the path:

$$d(i,j) = \sqrt{\sum_k (x_{i,k} - y_{j,k})^2} + \alpha \ , \qquad (1)$$

Secondly, as inspired by [7], with each new input frame we compute a backwards, offline DTW path starting from the current (t, h_t) position, over a square-shaped match cost matrix covering around 5 seconds in the immediate past. Since both dimensions are now "known", the alignment is considered to be more accurate, and we are able to compute the distance δ_t^{BACK} between coordinates where the main alignment path and the backwards, corrective path reach the border of the window. Whenever this deviation δ_t^{BACK} exceeds a threshold $\epsilon = 50$ms, we adjust the weighting coefficients to favour a path in the respective direction. Figure 1 illustrates this behaviour for two test sequences: one gradually slows down, then carries on at 80% tempo before abruptly reverting to the original speed, and the other takes the opposite direction. It becomes evident how the deviation between the main and the backwards path fosters the different tempo regimes.

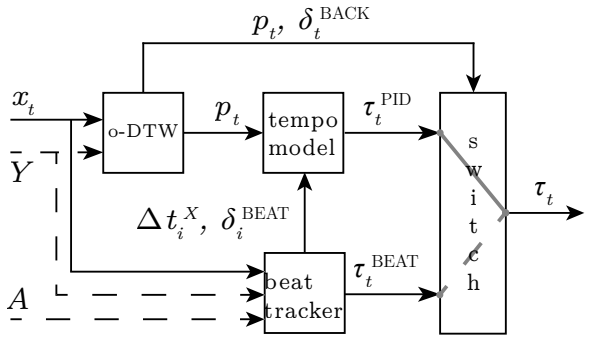


Figure 2: Framework architecture diagram. Dotted lines signify offline data pre-loading, regular arrows show online data flow. The beat tracker computes a tempo τ_t^{BEAT} based on the x_t input, and sends beat information to the alignment-based tempo model. The choice between τ_t^{BEAT} and τ_t^{PID} for tempo output is based on the alignment path p_t and its backwards error measure δ_t^{BACK}.

3.2 Tempo Model

In our system, for each time frame t the tempo model derives a relative tempo τ_t from the alignment path, so that:

$$\tilde{h}_t = \tilde{h}_{t-1} + \tau_t \ , \tag{2}$$

where $\tilde{h}_t \in \mathbb{Q}$ is the *accompaniment coordinate* in the reference series corresponding to the target frame t, and $\tilde{h}_0 = 0$.

Effectively, τ_t acts as a tempo *multiplier* in that it modulates the accompaniment playback speed. For each t, we define the *accompaniment error* ε_t as the difference between the alignment index and the accompaniment coordinate:

$$\varepsilon_t = h_t - \tilde{h}_t \ . \tag{3}$$

Our system contains four different tempo models, which the user can choose between. Hereon we describe and employ the *PID model*, inspired by the proportional-integral-derivative controller [17], which is a simple way to efficiently model adaptation and anticipation relative to trends in the alignment path. The model has the following output:

$$u_d(t) = K_P \varepsilon_t + K_I \sum_{i=0}^{t} \varepsilon_i + K_D \frac{\varepsilon_t - \varepsilon_{t-\Delta t}}{\Delta t} \ , \tag{4}$$

which produces the tempo multiplier:

$$\tau_t^{\text{PID}} = \frac{h_t - h_{t-\Delta t} + u_d(t)}{\Delta t} \ . \tag{5}$$

For the user-adjustable parameters, the Δt step has a default value of 500ms, and we found a good compromise between stability, response time and anticipation by setting: $K_P = 17 \times 10^{-3}$, $K_I = 3 \times 10^{-4}$, $K_D = 0.4$.

Finally, a *sensitivity* parameter S produces the threshold value ϵ^{S} which the accompaniment error ε_t needs to reach in order to activate the tempo model. For an appropriate sensitivity value, the system ignores minor fluctuations in the alignment path slope and holds the tempo steady at the last computed value. The threshold falls quadratically from one second to zero with the rise of sensitivity from 0 to 1:

$$\epsilon^{\text{S}} = (1 - S)^2 \times 100 \ . \tag{6}$$

The impact of the S setting is seen in Figures 4 and 6, where the time segments with a constant relative tempo have no background shading.

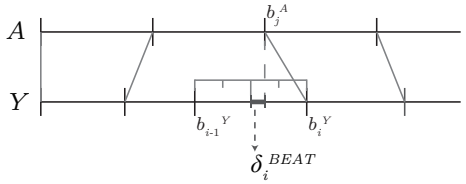


Figure 3: Computing the distance δ_i^{BEAT} between beats in the reference audio Y and their closest correspondents in the accompaniment track A. In this example, b_i^Y falls closer to an up-beat, so we translate it by $\frac{\Delta t_i^Y}{2}$ towards the closest upbeat b_j^A before measuring the distance.

4. BEAT TRACKING COMPONENT

We integrated the beat tracker in [5], which is publicly available [4] as a C++ class. The diagram in Figure 2 shows how the beat tracking module fits into the larger system framework.

We distinguish between the offline phase, where the reference audio to be matched and the accompaniment track are pre-loaded into the beat tracker and their beat positions marked as b_i^Y and, respectively, b_i^A, and the online phase, where the target audio beats b_i^X are detected in real time. Beat durations are measured as:

$$\Delta t_i^{[X,Y,A]} = b_i^{[X,Y,A]} - b_{i-1}^{[X,Y,A]} \ . \tag{7}$$

The Δt_i^X values replace the tempo update interval Δt from Equation (5) in real time, ensuring that new tempo values are computed by the PID tempo model with each new detected live beat.

Furthermore in the offline phase, for every reference beat b_i^Y, we compute the distance δ_i^{BEAT} to its corresponding accompaniment beat b_j^A. We take into consideration the possibility of the beats being in reverse phase, so if the distance is larger than a quarter of the current reference beat duration Δt_i^Y, then we translate b_i^Y by half of Δt_i^Y before again computing the distance:

$$\delta_i^{\text{BEAT}} = \min(|b_i^Y - b_j^A|, |b_i^Y \pm \frac{\Delta t_i^Y}{2} - b_j^A|) \ . \tag{8}$$

The actuation of the $\pm$ operation depends on whether b_i^Y, being the closest reference beat to b_j^A, comes before or after b_j^A. This process is depicted in Figure 3, where b_i^Y is translated backwards.

[4] See `https://github.com/adamstark/BTrack`.

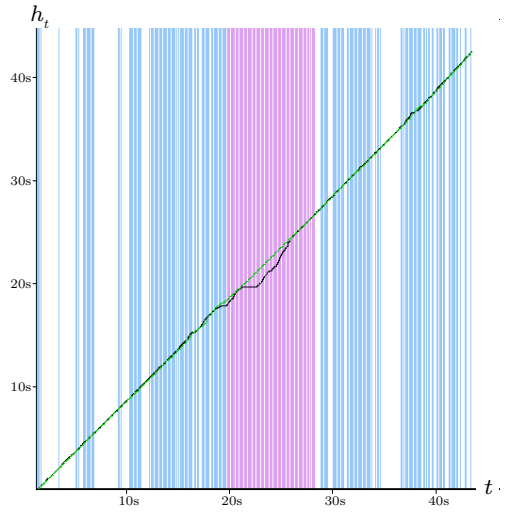


Figure 4: Online tempo tracking. The black line is the audio-to-audio alignment path; the green line traces the accompaniment coordinates produced by the system. Blue background shading marks time where τ_t^{PID} is in effect; purple marks τ_t^{BEAT} tempo; no background shading marks constant accompaniment tempo.

We use this reference-accompaniment beat distance δ_i^{BEAT} as a measure of how rhythmically "locked in" the reference part is to the backing track. This provides an indication of how closely we expect the live target to match the accompaniment beats, which makes it a good modulation factor for the tempo model's S sensitivity parameter. We rewrite Equation (6) as follows:

$$\epsilon_i^{\text{S}} = (1 - S)^2 \times 100 + \delta_i^{\text{BEAT}} \quad . \tag{9}$$

Now for each new b_i^Y beat reached, the tempo model's activation threshold moves depending on how tightly we expect the target to match the reference. Thus, for "loose" beats we raise the threshold, meaning that small deviations are ignored by the accompaniment and the tempo is kept constant. All the examples in this paper were realised using $S = 1$, which would have produced a global zero threshold under Equation (6).

The beat tracking module derives a local tempo BPM estimate from the observed beat durations [5]. To produce the relative tempo τ_t^{BEAT} that can drive the accompaniment, we divide the live tempo estimate by the tempo detected at the same beat in the accompaniment track. We define two situations where the system's relative tempo output switches from the alignment-based value τ_t^{PID} to the beat-based one τ_t^{BEAT}:

1. if the tempo produced is more than 3 times slower or faster than the reference, or

2. if the backwards DTW deviation δ_t^{BACK} exceeds a threshold $\epsilon^{\text{B}} = 174$ms, and the difference between the alignment slope [5] $\frac{dh}{dt}$ and τ_t^{BEAT} is greater than $\epsilon^{\text{T}} = 0.15$.

[5] smoothed over the last 40 frames, or 464ms.

Algorithm 1 Switching between the alignment-based tempo model τ_t^{PID} and the beat-based one τ_t^{BEAT}.

$calc \leftarrow$ NONE
$waiting \leftarrow 0$
$\tau_0 \leftarrow 1$
for all frames x_t in X **do**
  if $\varepsilon_t > \epsilon_i^{\text{S}}$ **then**
    if $(\delta_t^{\text{BACK}} > \epsilon^{\text{B}}$ **and** $|\frac{dh}{dt} - \tau_t^{\text{BEAT}}| > \epsilon^{\text{T}})$ **or**
    $(\tau_{t-1} \notin (\frac{1}{3}, 3))$ **then**
      $waiting \leftarrow \Delta t_i^A$
    end if
    if $waiting > 0$ **then**
      $\tau_t \leftarrow \tau_t^{\text{BEAT}}$ {*computed by the beat tracker.*}
      $calc \leftarrow$ BEAT
    else
      if $calc =$ BEAT **then**
        $\tilde{h}_t \leftarrow h_t$ {*jump back to alignment path pos.*}
      end if
      $\tau_t \leftarrow \tau_t^{\text{PID}}$ {*computed in Equation (5).*}
      $calc \leftarrow$ PID
    end if
  else if $\varepsilon_t \leq 1$ **and** $calc \neq$ NONE **then**
    $\tau_t \leftarrow \frac{dh}{dt}$ {*use current path slope.*}
    $calc \leftarrow$ NONE {*disable tempo model.*}
  end if
  if $waiting > 0$ **then**
    $waiting \leftarrow waiting - 1$
  end if
  $\tilde{h}_t \leftarrow \tilde{h}_{t-1} + \tau_t$
end for

When one of the two conditions is hit, the beat tracker drives the tempo for at least one Δt_i^A beat. Afterwards, if the conditions are inactive, the alignment-based tempo model regains control, and the accompaniment cursor $\tilde{h}_t$ jumps to the respective path position h_t. This entire switching algorithm is laid out in Listing 1.

We can follow the algorithm's execution through the example shown in Figure 4. For the first third of the run-through, the live target closely matches the reference. In the next third, the musician starts improvising, keeping the same tempo but diverging from the original pitches significantly. The $(\delta_t^{\text{BACK}} > \epsilon^{\text{B}}$ **and** $|\frac{dh}{dt} - \tau_t^{\text{BEAT}}| > \epsilon^{\text{T}})$ condition is activated and the tempo is now controlled by the beat tracker, $\tau_t \leftarrow \tau_t^{\text{BEAT}}$. The condition remains active until a few seconds after the musician has resumed playing the scored pitches. By that time, the online DTW path has rejoined the accompaniment path, so the transition back to the original tempo model is seamless.

In the next section we study a more challenging case and test the limits of our proposed accompaniment system.

5. CASE STUDY

The alignment performance of our online DTW engine has been evaluated in [3], with results on par with equivalent implementations. The beat tracker is evaluated in [5]. In order to thoroughly assess the performance of models presented in this paper, we would require experimental pro-

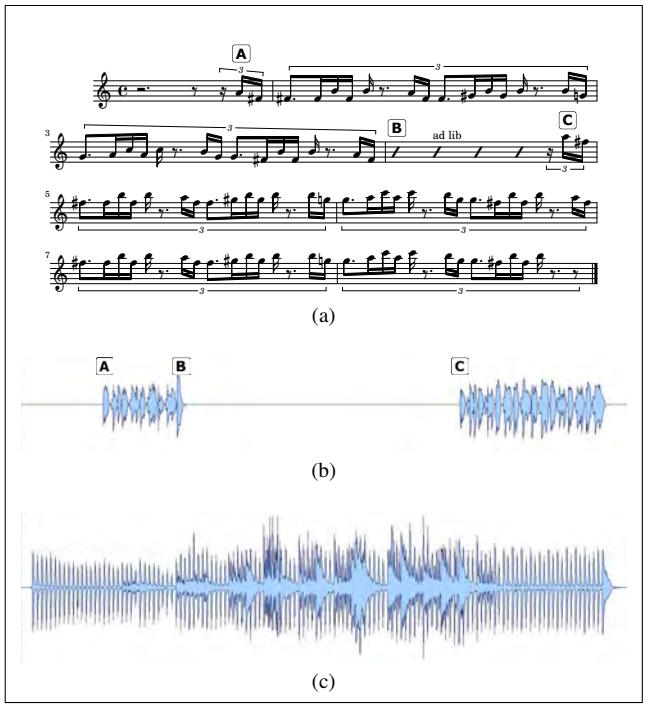


Figure 5: Example scenario. (a) lead instrument score, unknown by the system; (b) reference track; (c) accompaniment track. (b) and (c) are pre-loaded into the system.

cedures and reference benchmarks that exceed the current standard methods for score following or beat tracking tasks, which mostly measure keyframe-matching ability without specifically studying the musical quality of the resulting accompaniment.

The case for a wider test bench that includes not just several performances of a piece, but also the corresponding accompaniment tracks, is further strengthened by the beat-based tempo model we introduced in section 4, which strongly relies on backing track information. In the absence of such an evaluation database, a preliminary case study will demonstrate the qualitative improvements over our previous system and equivalent followers.

The materials we produced for this example scenario are presented in Figure 5. A backing track (drums) plays by itself for one measure to cue in the lead instrument (guitar). The two play together for one measure (section A), followed by an improvisation (section B) where the backing track keeps a steady beat. The lead instrument decides when to conclude the improvisation by entering the final two measures (section C).

There are two major questions that our system must answer, without referring to any symbolic information or programmed instructions: what to do when the musician starts improvising, and how to latch back on at the conclusion.

Figure 6 shows one realisation of the scenario, where the musician first misses the cue to start playing, which causes the alignment to remain stuck at the start of section A. Other online aligners might produce the same result, but our machine does not stop here: the beat tracker takes control, setting the tempo to 1, which allows the *waiting* variable to elapse during one beat. At the end of this beat, the accompaniment cursor goes back to the baseline, and the process is repeated. Thus, our musician (and the audience) is given a steady beat; once they start playing over it, the alignment-based tempo model regains control.

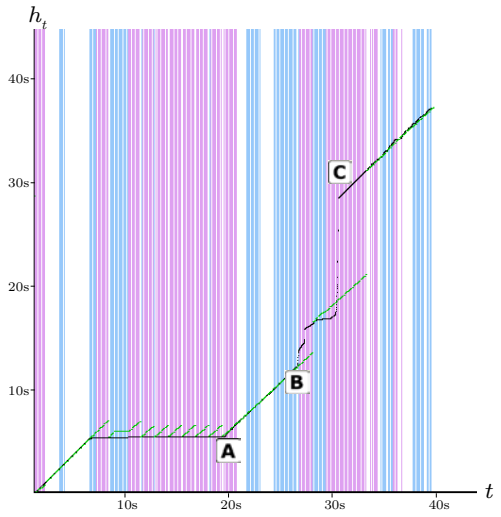


Figure 6: Example execution of the scenario in Figure 5. Notations in Figure 4 apply. See text for interpretation.

To model the space for improvisation, we found that filling the space in the reference audio with musical silence (actually the natural background noise of the instrument) gives the desired result. This way, the contrast between (target) activity and (reference) silence makes it instantly obvious when the improvisation begins: the online DTW starts evolving unpredictably, δ_t^{BACK} explodes and the beat tracker takes control.

The realignment in the final measures (as seen in the vertical jump in Figure 6) depends on one important condition: that the musician pause between the end of the improvisation and the start of the scored coda. This allows the DTW process to match one transition to the other. We found for our particular example [6] a pause of 1.1s to be sufficient, for an online DTW window of 1.49s [7].

6. CONCLUSIONS

We have presented an online audio-to-audio following and accompaniment system that combines a tempo model based on the alignment path produced by an online DTW process with a tempo model tied to beat tracking, without reference to any symbolic score data. The two models constantly inform each other, producing a synergistic relationship.

This framework is geared to a specific scenario, where a musician creates a reference audio track by playing along to a fixed backing track. This accompaniment track is then warped in real time to match a live target. Such a scenario applies more to popular beat-based music genres than the classical concerto or solo performances that followers such as [2,18] are designed around. Thus, while the added rigid-

[6] All test and demo files for this paper are available at `https://github.com/RVirmoors/RVdtw-/tree/master/_smc`

[7] $c = 128$ frames with a hop of 512 samples, at 44.1kHz sample rate.

ity serves beat-centred material well, it is less appropriate for strong rubato, where pure alignment-based models perform better. The user can address this by raising the ϵ^{B} and ϵ^{T} thresholds, causing the beat tracker to engage less easily.

We anticipate several avenues for future work. Firstly, an evaluation framework based on a cross-genre database of backing tracks and isolated performances is needed in order to ensure measurable progress. Our proposed embedding of beat tracking into the alignment process is certainly not the only possible method, and so far we have relied on piecewise experimentation to move forward.

Secondly, we are looking to develop new tempo models that make integral use of both performance alignment and beat tracking. The adaptation of the PID model introduced in section 4 is a start, but we might conceive models from the ground up with this configuration in mind.

Thirdly, we intend to further increase system robustness through parallel observations. Among the research directions worth considering are multi-agent following [19], asynchrony compensation [20] to capture the timing nuances of several musical facets, and parallel trackers for two or more musicians jointly driving the accompaniment.

Finally, we plan to move beyond warped backing tracks, to produce more dynamic accompaniments where the timing information extracted from live target performance(s) informs temporal deviations within the accompaniment's individual components.

7. REFERENCES

[1] S. Dixon, "Live tracking of musical performances using on-line time warping," in *International Conference on Digital Audio Effects (DAFx)*, 2005, pp. 92–97.

[2] A. Arzt and G. Widmer, "Simple tempo models for real-time music tracking," in *Sound and Music Computing conference (SMC)*, 2010.

[3] G. Burloiu, "An online audio alignment tool for live musical performance," in *Electronics and Telecommunications (ISETC)*, Nov 2014.

[4] A. Robertson and M. Plumbley, "B-keeper: A beat-tracker for live performance," in *International Conference on New interfaces for musical expression (NIME)*. ACM, 2007, pp. 234–237.

[5] A. M. Stark, M. E. Davies, and M. D. Plumbley, "Real-time beat-synchronous analysis of musical audio," in *International Conference on Digital Audio Effects (DAFx)*, 2009, pp. 299–304.

[6] C. Fremerey, M. Müller, and M. Clausen, "Handling repeats and jumps in score-performance synchronization." in *International Society for Music Information Retrieval conference (ISMIR)*, 2010, pp. 243–248.

[7] A. Arzt, G. Widmer, and S. Dixon, "Automatic page turning for musicians via real-time machine listening." in *European Conference on Artificial Intelligence (ECAI)*, 2008, pp. 241–245.

[8] A. Cont, "On the creative use of score following and its impact on research," in *Sound and Music Computing conference (SMC)*, Jul. 2011.

[9] M. Muller and D. Appelt, "Path-constrained partial music synchronization," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, March 2008, pp. 65–68.

[10] M. Grachten, M. Gasser, A. Arzt, and G. Widmer, "Automatic alignment of music performances with structural differences," in *International Society for Music Information Retrieval conference (ISMIR)*, November 2013.

[11] R. Macrae and S. Dixon, "Accurate real-time windowed time warping," in *International Society for Music Information Retrieval conference (ISMIR)*, 2010, pp. 423–428.

[12] N. Montecchio and A. Cont, "A unified approach to real time audio-to-score and audio-to-audio alignment using sequential montecarlo inference techniques," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2011, pp. 193–196.

[13] B. Xiong and O. Izmirli, "Audio-to-audio alignment using particle filters to handle small and large scale performance discrepancies," in *International Computer Music Conference (ICMC)*, 2012.

[14] A. M. Stark and M. D. Plumbley, "Performance following: Real-time prediction of musical sequences without a score," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 20, no. 1, pp. 190–199, 2012.

[15] P. Toiviainen, "Real-time recognition of improvisations with adaptive oscillators and a recursive bayesian classifier," *Journal of New Music Research*, vol. 30, no. 2, pp. 137–147, 2001.

[16] A. M. Stark and M. D. Plumbley, "Real-time chord recognition for live performance," in *International Computer Music Conference (ICMC)*, 2009.

[17] G. F. Franklin, J. D. Powell, and M. L. Workman, *Digital control of dynamic systems*. Menlo Park: Addison-Wesley, 1998, vol. 3.

[18] C. Raphael, "Music plus one and machine learning," in *International Conference on Machine Learning (ICML)*, 2010.

[19] A. Arzt and G. Widmer, "Real-time music tracking using multiple performances as a reference," in *International Society for Music Information Retrieval conference (ISMIR)*, 2015.

[20] S. Wang, S. Ewert, and S. Dixon, "Compensating for asynchronies between musical voices in score-performance alignment," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, April 2015, pp. 589–593.

FACTORSYNTH: A MAX TOOL FOR SOUND ANALYSIS AND RESYNTHESIS BASED ON MATRIX FACTORIZATION

Juan José Burred
Paris, France
jjburred@jjburred.com

ABSTRACT

Factorsynth is a new software tool, developed in the Max environment, that implements sound processing based on matrix factorization techniques. In particular, Non-negative Matrix Factorization is applied to the input sounds, which produces a set of temporal and spectral components that can be then freely manipulated and combined to produce new sounds. Based on a simple graphical interface that visualizes the factorization output, Factorsynth aims at bringing the ideas of matrix factorization to a wider audience of composers and sound designers.

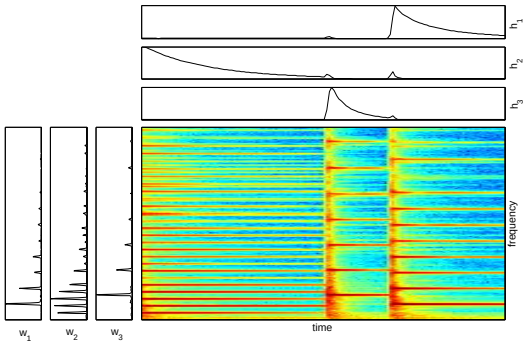
1. INTRODUCTION

Any kind of data in matrix form can be subjected to factorization, i.e., to an algorithm that yields two or more output matrices (the factors) which, when multiplied back together, produce an approximation of the input. There is a wide range of factorization algorithms that can produce very different factor matrices, depending on the constraints imposed by the desired application. By analyzing the resulting factor matrices it is possible to discover and separate important underlying components, often called *latent variables*, that were hidden and mixed within the original data. Because of this, factorization is central to many computing fields such as data compression, computer vision or machine learning.

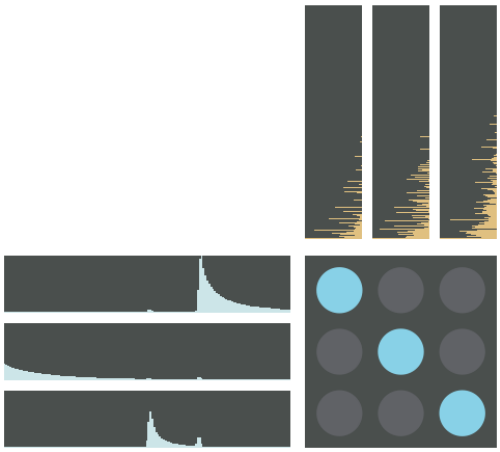
In the audio domain, matrix factorization is most often applied to the magnitude or power spectrogram, which is a matrix whose rows are time-varying energies of individual frequency bands, and whose columns are spectra at given times. One of the most widely used factorization algorithms in sound applications is Non-negative Matrix Factorization (NMF) [1], which imposes the constraint that all elements of the input and output matrices have to be zero or positive. NMF results in two factor matrices, one containing spectra (called *bases*) and the other containing temporal functions (called *activations*). The combination of one of the spectral bases with its corresponding activation results in a *component sound*, an approximation to a sonic entity or event contained in the input signal. Component sounds can be, for instance, individual notes, drum hits, or any kind of temporally or spectrally distinctive event.

NMF is thus ideal for applications that require analyzing or resynthesizing separate elements of the input sound, and is currently widely used in fields such as source separation or music information retrieval. An illustration of a simple sound decomposed by NMF is shown in Fig.1(a). The input sound (spectrogram shown) is a sequence of 3 piano notes of different pitches. The output of the factorization are a set of 3 spectral bases (displayed at the left of the spectrogram) and a set of temporal activations (displayed on top). It can be seen that the peaks in the activations correspond to the temporal positions of the corresponding spectral bases in the input sound.

The application of matrix factorization to musical cre-



(a) Input spectrogram with extracted spectral bases (left) and activations (above)



(b) Display convention used in Factorsynth

Figure 1. Visualization of a simple sound (three-note piano melody) decomposed by NMF.

ation is fairly recent [2–5]. As a sound decomposition method, it fits well into the analysis/resynthesis paradigm of computer and electronic music, in which a sound is modified by manipulating parameters resulting from its previous analysis (or, in cross-synthesis, from the analysis of a second sound). In traditional additive analysis/resynthesis (phase vocoder), the decomposition bases are sinusoids. In this context, matrix factorization can be seen as a higher-abstraction version of additive analysis/resynthesis in which each sinusoid has been replaced by a full spectrum.

A framework for performing sound modifications and cross-synthesis based on factorization was presented in [6]. This article presents a graphical implementation thereof in the Max environment. The basic principle of Factorsynth is the ability to freely recombine any spectral basis with any temporal activation resulting from factorization. In other systems aimed at analysis and separation, each basis always remains coupled with its corresponding activation, since the goal is to reconstruct elements that are actually present in the original sound. In contrast, here the goal is to create new sounds, and so arbitrary recombinations are allowed.

A preliminary implementation was presented in the form of a command-line executable [6], which was of limited usability and control capabilities. The new Max version provides a graphical interface and thus the possibility for the user to visualize the result of factorization, listen to the separated components, edit the extracted bases and activations, and closely control the resynthesis process. The ability to freely perform any base/activation combination is emphasized by the central element of the graphical interface: a switchboard that symbolizes the couplings between bases and activations. An example of the Factorsynth visualization of factorization and recombination is shown in Fig.1(b), which corresponds to the same three-note piano sound of Fig.1(a). For easier alignment, the spectral bases are displayed above the switchboard, and the temporal activations to its left. An activated button on the switchboard means that the basis above it and the activation to its left are to be combined for resynthesis. In the figure example, the switchboard has its diagonal elements activated, which means that in this case resynthesis will approximate the original sound without modifications.

Factorsynth is freely available for download [1]. Several sound examples are also presented in the download page.

2. THE FACTORSYNTH INTERFACE

A number of controls are available in the interface, together with the display of the components and of the switchboard. Two usage scenarios will be considered here: the manipulation of a single sound and cross-synthesis.

2.1 Single-sound manipulation

Fig.2 shows the Factorsynth interface in a single-sound manipulation scenario, running on Max 7. Note that the

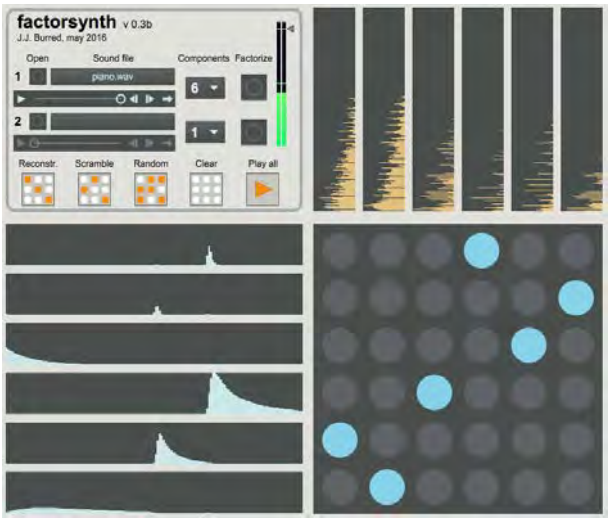


Figure 2. Main interface of Factorsynth for processing of a single sound.

interface allows to load two files; for single-sound processing, only the first file is loaded. Next to the file name is a menu to select the number of components for the extraction. Remember that one component corresponds to one base/activation pair. In Fig.2, six components have been set for factorization.

When the number of components is chosen, the size of the switchboard and the number of display areas for the bases and activations are adjusted. NMF decomposition is launched when clicking on the 'factorize' button corresponding to the loaded sound. Computation time is around 25% of the length of the input file (a 4s file will take 1s to decompose). After computation, the display areas are filled with the bases (above) and the activations (at the left). The bases are displayed in logarithmic amplitude and linear frequency.

Clicking the 'factorize' button again repeats factorization. Successive factorization runs can produce slightly different results since NMF is a numerical optimization algorithm that relies on random initialization [2]. This can result in small amplitude differences and, more noticeably, a different ordering of the output bases and activations.

There is a scale ambiguity of the factors produced by any factorization, since a product $(cx) \times (y/c)$ is the same for any value of c. In other words, it is possible to arbitrarily transfer energy from the bases to the activations, or vice-versa, without changing the validity of the factorization. In Factorsynth, the following convention has been applied:

1. First, the spectral bases are *individually* max-normalized (i.e., re-scaled so that they all reach the maximum amplitude of the display area) and the resulting energy differences transferred to the activations.

2. Then, for display, the activations are *globally* max-normalized (i.e., re-scaled so that only one reaches the maximum amplitude).

[1] `http://www.jjburred.com/software/factorsynth`

[2] The interested reader can find details about the NMF algorithm in the extensive literature (e.g.: [7]).

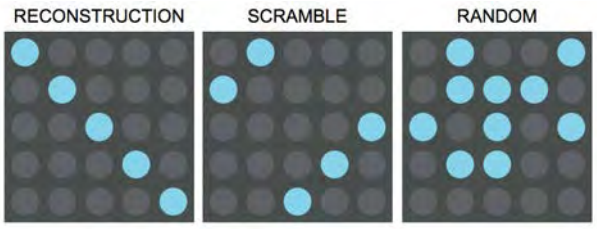


Figure 3. Modes for automatic switchboard configuration.

What this means in terms of interpretation of the graphical output is that the energy information is contained in the activations. A low-energy component will have a low-amplitude activation (such as the last component at the bottom of Fig.2), but its spectral base will still span the whole display range.

The bases and activations are displayed on editable multislider objects, so that the user can draw on them to modify the sound to be resynthesized.

The user can then click on the switchboard buttons to assign the desired base/activation pairs for the resynthesis. When clicking on a switchboard button, a resynthesis and playback of the corresponding base/activation pair is instantly launched (the computation time needed for resynthesis is negligible) in order to listen to that separate component. Thus, buttons on the diagonal will play coupled base/activation pairs which were present in the original sound, and off-diagonal buttons will generate artificial components not originally present. Once the desired individual connections have been made, the full resynthesized sound can be played by clicking on the 'play all' button.

Instead of manually selecting the switchboard connections, there are 4 buttons to set them up automatically (see Fig.3):

- **Reconstruction**. Sets the diagonal buttons on, all the others off. When full resynthesis is performed, this results in the playback of an approximation of the original sound. Reconstruction is never identical to the input, since NMF, like most factorization algorithms, is approximate.
- **Scramble**. Generates a random permutation of the connections. The connections are one-to-one (injective).
- **Random**. All connections are randomly chosen. Repetitions are possible: a single activation can control several bases, or several activations can control a single base.
- **Clear**. Sets all connections to zero.

2.2 Cross-synthesis

When two input sound files are selected, the interface enters in cross-synthesis mode (Fig.4). The switchboard is divided into four parts. The two parts on the diagonal (with the blue buttons) correspond to the single-sound manipulation connections, controlling the base/activation combinations within each of the input sounds. The two other

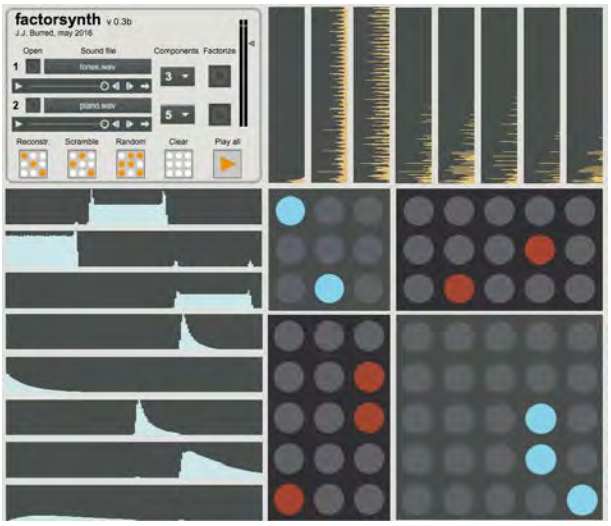


Figure 4. Factorsynth interface in cross-synthesis mode.

sectors of the switchboard, distinguished by their red buttons, control connections between bases of one sound and activations of the other, thus generating *cross-components*.

In this type of factorization-based cross-synthesis [5], internal temporal elements of one sound can thus control internal spectral shapes of the other.

3. RESYNTHESIS

It is worth going into some detail about the resynthesis process in order to understand the sonic results of Factorsynth. As its name implies, NMF works only on real, non-negative numbers, which means that phase information is discarded and only magnitude or power spectrograms are taken as the input. The combination of bases and activations (also comprised of real numbers) produce a set of magnitude spectrograms from which the synthesized output sounds have to be generated. Since the phase information was discarded from the outset, there are two options at this point:

- Either new phase information is generated randomly or by means of an optimization method, such as the Griffin and Lim algorithm [8], or
- Phase information is taken from the original input complex spectrogram.

The second option has been chosen for Factorsynth due to its superior sound quality and faster computation time. However, instead of directly attaching the input phases to the output spectrogram, Factorsynth uses Wiener filtering [6], which is known from source separation to produce more natural sounds.

Wiener filtering consists of computing a time-frequency mask from the output magnitude spectrograms that, when applied (by element-wise multiplication) to the input complex spectrogram produces the output spectrogram. Such a Wiener mask can be understood as a time-varying filter that is, in effect, performing subtractive synthesis from the input sound. Once the output complex spectrogram has been

obtained in this way, an overlap-add algorithm is applied to invert it and produce the output time-domain signal.

The choice of Wiener filtering for resynthesis has an important implication for Factorsynth: if a high-energy activation is combined with an originally unrelated basis, it can happen that the resulting component will nevertheless be of low energy. Indeed, frequency contents can be hardly amplified if there is only little energy at those frequencies in the corresponding position of the input sound.

Factorsynth is able to handle both mono and stereo signals. For stereo signals, NMF is applied to the sum of both channels, and the resulting time-frequency masks are applied to both left and right input spectrograms to generate each of the output channels.

4. THE FACTORSYNTH~ EXTERNAL

The core of the Factorsynth Max patch is the factorsynth~ external object. It implements both NMF decomposition and Wiener resynthesis. Each factorsynth~ object handles a single input file, so for cross-synthesis, two instances are needed. Linear algebra operations inside the object (FFTs, matrix multiplications, outer products...) are implemented using Apple's highly optimized vDSP library, part of the Accelerate framework.

A usage example for both factorization and resynthesis is shown in Fig.5.

4.1 Factorization operation

The sequence of operations needed to perform a factorization is the following:

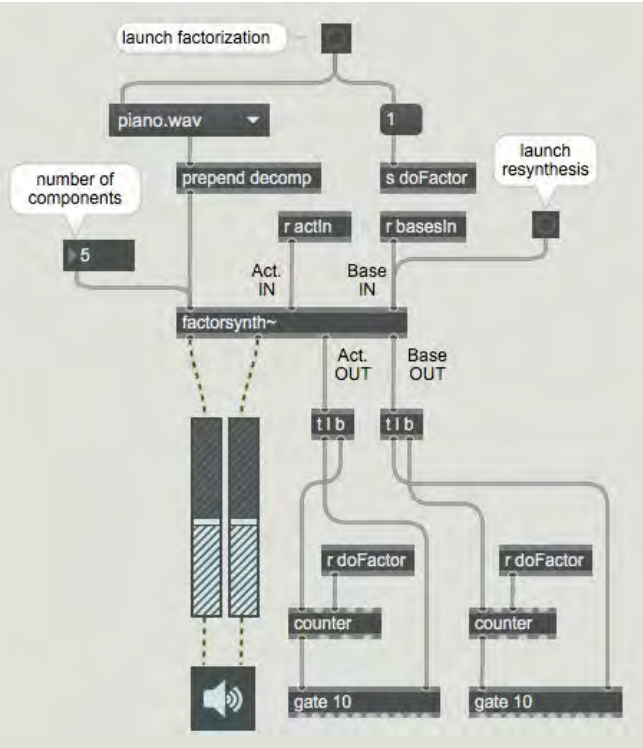


Figure 5. Usage example of the `factorsynth~` external.

1. The number of components K is passed as an integer to the left inlet.

2. A message of the form `decomp filename` is passed to the left inlet, launching spectrogram computation and NMF factorization. The specific NMF algorithm implemented in factorsynth~ is Kullback-Leibler (KL) NMF.

3. Activations are output from the third outlet as a sequence of K lists.

4. Bases are output from the fourth outlet as a sequence of K lists.

The output list sequences could then, for instance, be handled by gate objects to be sent to separate display areas, as shown in Fig.5.

4.2 Resynthesis operation

To launch a resynthesis with a given set of base/activations pairs, the following operation sequence must be performed:

1. A sequence of lists are read into the middle inlet, containing the activations.

2. A sequence of lists are read into the right inlet, containing the bases.

3. A bang is sent to the right inlet, signaling the end of the incoming data and launching the computation of the global Wiener mask and its application to the input spectrogram. There is no need to reload the input audio file since the spectrograms are stored in memory for every instance of a factorsynth~ object.

4. The masked spectrogram is converted back into the time domain using the overlap-add technique. The resulting audio is output as a mono signal from the first outlet, or as a stereo signal from the left and second outlets.

5. FUTURE DEVELOPMENTS

Aside from being computationally intensive (as mentioned before, around 25% of the input file length), NMF factorization is an intrinsically off-line operation, since the full length of the input signal has to be observed prior to starting the decomposition algorithm. Thus, the current version of Factorsynth is non-real-time and works only on sound files. An important goal for future versions is the ability to process incoming audio data in real- or near-real-time.

A relatively straightforward way of implementing a real-time cross-synthesis would be to perform a preliminary factorization of a sound and then use an arbitrary selection of its stored spectral bases to filter the incoming audio stream. A second, most sophisticated way would be to explore online factorization algorithms [9] and assess the feasibility of a quick decomposition of the input stream.

Another direction for future developments will be the exploration of alternative interfaces for the representation and recombination of the extracted bases and activations. The

current interface, based on displaying individual bases and activations, might become ineffective when using a large number of components (in source separation, tens of components are often used). In that case, an interface based on a 2-D scatter plot might be more appropriate, in which bases or activations could be represented as points and placed in coordinates corresponding to a given spectral or temporal shape feature. Connections in the cross-synthesis switchboard could then be generated automatically following criteria of proximity in such a feature space.

6. CONCLUSIONS

This paper has introduced Factorsynth, a graphical tool for the Max environment that exploits matrix factorization techniques to perform sound manipulations. Stemming from data analysis and machine learning, matrix factorization techniques remain relatively unknown in the field of computer music. It has recently been shown that such techniques constitute a promising new alternative to sinusoidal or source/filter models for analysis/resynthesis applications, and they allow a new kind of cross-synthesis that operates at the level of internal elements of the involved sounds (spectral shapes, salient temporal events...), rather than on global features. Factorsynth aims at bringing those new concepts to a wider audience of composers and sound designers. Its simple graphical interface visualizes all extracted elements and allows the user to modify them and carefully control their combination before resynthesis.

Acknowledgments

The author would like to thank Marco Liuni, Emanuele Palumbo, Carmine Emanuele Cella and Nicolas Obin for the insightful discussions and helpful suggestions.

7. REFERENCES

[1] D. D. Lee and H. S. Seung, "Learning the parts of objects by non-negative matrix factorization," *Nature*, vol. 401, pp. 788–791, 1999.

[2] S. S. Topel and M. A. Casey, "Elementary sources: Latent component analysis for music composition," in *Proc. ISMIR*, Miami, USA, 2011.

[3] R. Sarver and A. Klapuri, "Application of non-negative matrix factorization to signal-adaptive audio effects." in *Proc. DAFX*, Paris, France, 2011.

[4] R. Maguire, "Creating musical structure from the temporal dynamics of soundscapes," in *Proc. Int. Conf. on Information Sciences, Signal Processing and Applications (ISSPA)*, Montreal, Canada, 2012.

[5] J. J. Burred, "Cross-synthesis based on spectrogram factorization," in *Proc. ICMC*, Perth, Australia, 2013.

[6] ——, "A framework for music analysis/resynthsis based on matrix factorization," in *Proc. ICMC*, Athens, Greece, 2014.

[7] D. Lee and H. Seung, "Algorithms for non-negative matrix factorization," in *Neural Information Processing Systems*, Denver, USA, 2001.

[8] N. Sturmel and L. Daudet, "Signal reconstruction from STFT magnitude: a state of the art." in *Proc. DAFX*, Paris, France, 2011.

[9] A. Lefévre, F. Bach, and C. Févotte, "Online algorithms for nonnegative matrix factorization with the Itakura-Saito divergence," in *Proc. WASPAA*, New Paltz, USA, 2011.

VR 'SPACE OPERA': MIMETIC SPECTRALISM IN AN IMMERSIVE STARLIGHT AUDIFICATION SYSTEM

Benedict Carey
Hochschule für Musik und Theater Hamburg
benedict.carey@hfmt-hamburg.de

Burak Ulaş
Izmir Turk College Planetarium
bulash@gmail.com

ABSTRACT

This paper describes a system designed as part of an interactive VR opera, which immerses a real-time composer and an audience (via a network) in the historical location of Göbeklitepe, in southern Turkey during an imaginary scenario set in the Pre-Pottery Neolithic period (8500-5500 BCE), viewed by some to be the earliest example of a temple, or observatory. In this scene music is generated, where the harmonic material is determined based on observations of light variation from pulsating stars, that would have theoretically been overhead on the 1st of October 8000 BC at 23:00 and animal calls based on the reliefs in the temple. Based on theoretical observations of the stars V465 Per, HD 217860, 16 Lac, BG CVn, KIC 6382916 and KIC6462033, frequency collections were derived and applied to the generation of musical sound and notation sequences within a custom VR environment using a novel method incorporating spectralist techniques. Parameters controlling this 'resynthesis' can be manipulated by the performer using a Leap Motion controller and Oculus Rift HMD, yielding both sonic and visual results in the environment. The final opera is to be viewed via Google Cardboard and delivered over the Internet. This entire process aims to pose questions about real-time composition through time distortion and invoke a sense of wonder and meaningfulness through a ritualistic experience.

1. INTRODUCTION

The system we have developed forms the basis for the forthcoming opera *Motherese*. It immerses a real-time composer and an audience (via a network) in the historical location of Göbeklitepe, in southern Turkey during an imaginary scenario set in the Pre-Pottery Neolithic period (8500-5500 BCE). A description of the networking features of this package is beyond the scope of this paper, instead we will concentrate on the virtual staging, sound resynthesis and sonification aspects of out system.

Rather surprisingly so, FFT analysis, which is so commonly employed in the composition of spectral music, originates from a formula designed for rapidly calculating the elliptical orbits of planetary bodies. This early version of the DFT is a development attributed to Alexis-Claude Clairaut in 1754 [1], but one could look even further back to ancient Babylonian mathematics if the term 'spectral analysis', which is often used to describe the method by which spectralist composers derive musical material for their compositions, and is sometimes a stand in for 'harmonic analysis' [2]. Of course since the term harmonic analysis already connoted something entirely different amongst musicologists by the time the French Spectralist tradition began in the 1970's, this linguistic evolution makes sense, despite being a slightly confusing side effect both of the difficulties of categorization and the interdisciplinary nature of Spectralism. Mostly the term spectral analysis is used in an even more narrow sense when speaking in the context of spectral music however, to refer to DFT or FFT analysis of audio signals containing content from within the audible frequency range (20 Hz and 20,000 Hz) to yield a collection of frequencies (pitches) and their amplitude (dynamic) variance over time for a composition. Indeed the stipulation that spectral analysis produces musical results is a creative leap of faith that supports the co-option of this process into the composer's repertoire of compositional techniques, and for good reason. Why shouldn't one look to mathematics to help build a stronger understanding of music via recognition of the structural underpinnings of sound, the very concrete from which this art form emerges?

Yet at the same time why stop at the analysis of sound to produce frequency collections from which to derive new harmonies and timbres? FFT analysis is a tried and tested tool for modelling a musical representation of a subject, using the program *Macaque*[1] in combination with a SDIF file for example, one can easily track the movement of a sound spectrum over time such as was done by Gérard Grisey through a similar method for his seminal work *Partiels* [3], whose methods we will focus on here. If FFT analysis translates its usefulness so well from the realm of the cosmos into such a diverse array of phenomena such as audio signal processing, medical imaging, image processing, pattern recognition, computational chemistry, error correcting codes, and spectral methods for PDEs [4], it is perhaps no more worthy a candidate for the source of frequency based musical inspiration than any other similar method of observing the natural world's many oscillations.

So is the practice of using other algorithmic methods to interpret natural phenomena any less valid or useful to the composer? The process of sonification, or

[1] See http://georghajdu.de/6-2/macaque/

audification as it is more commonly known to astronomers, is fairly widespread due to the pragmatic consequence, of speeding up time-consuming manual data analysis. When approaching spectral music composition in real-time scenarios as is the case in the project presented in this paper, the speed at which abstractions of these forms can be realised as sound is paramount to their success as music of course, but perhaps the most important aspect is the representation of the entity in music, an entity which itself does not transmit any sound through the great vacuum of space, over distances of multiple light years. It is therefore fairly reasonable to assume that the usefulness of spectral compositional methods remains, even if FFT analysis or some tonal system built around the 'natural harmonic series' is removed from this linear process, and replaced with another algorithm designed to derive a similar kind of 'tonal reservoir' [2] for our purposes.

The use of starlight audification to create musical textures has precedent [5] but has so far not been incorporated into a real-time spectral composition system. Of primary relevance to this particular research project is the clear, discernable embodiment of extra-musical objects inside of a musical context known as 'Mimetic Spectralism' [6]. It may therefore prove no more relevant to us to base a composition on 'sound' itself, once it is abstracted to the point of mathematical analysis, than on any other method of analysis of a physical phenomenon, which we consider a form of embodiment. The apotheosis of sound as a kind of 'living object with a birth, lifetime and death' [11] as Grisey put it, is not the focus here. Certainly in the light of careful review by a skilled composer (or just one with the right software tools), any collection of frequencies can be stretched through a wide array of aesthetic extremes, as the practice of spectralism is after all an impressionistic exercise [7].

2. SPECTRALISM AND BELIEF

It has been observed that the use of FFT analysis in music composition may imply an extra-musical dimension to the piece concerned [8] The assertion that what the composer produces using spectral techniques is music often comes along with certain presumptions and philosophies about the nature of sound and music perception. Inevitably, this extra-musical motivation pushes this music into the territory of referential expressionism [9]. One tendency among proponents of spectralism is to justify their use of spectral technique by referencing its links to the sciences. Many proponents of the movement claim that forms extracted from within sonic events represent a natural and fundamental order of music as evidenced by the micro-structure of sound. Despite the scientific origins of the techniques used in spectral composition, they are of course not by themselves scientific proofs of 'musical truths'. Instead, it has instead been suggested that when 'new art' is generated from the analysis of 'natural' objects, this indicates naturalism as the philosophical basis for the art piece concerned [8]. Extra-musical representation in spectral music is not always the intention of the composer, but sometimes it is unavoidable. Gérard Grisey exhibits a kind of devotional respect for sound that is almost animistic. Grisey proposed that spectral music reminds the listener that sound is in fact living.

"Spectralism is not a system. It's not a system like serial music or even tonal music. It's an attitude. It considers sounds, not as dead objects that you can easily and arbitrarily permutate in all directions, but as being like living objects with a birth, lifetime and death." (Grisey, 1996)

With this anthropomorphic approach to sound Grisey displays this reverence towards the source of his compositions and his muse, sound itself. In the same interview he mentions that while his music represents a 'state of sound' it is simultaneously a discourse:

"I would tend to divide music very roughly into two categories. One is music that involves declamation, rhetoric, language. A music of discourse... The second is music which is more a state of sound than a discourse... And I belong to that also. I would put myself in this group. Maybe I am both, I don't know. But I never think of music in terms of declamation and rhetoric and language." (Grisey, 1996)

With this impetus we created a system that explores animism and discourse through re-synthesis in a ritualistic setting. We set out to create a music of discourse, which embodies starlight and animal calls in the music, which is generated via commonly used spectralist techniques such as Orchestral resynthesis and 'spectra as reservoir' among others. Here we will detail our approach to 'Mimetic Spectralism'.

3. THEORETICAL FRAMEWORK FOR STARLIGHT AUDIFICATION

Pulsating stars simply expand and shrink within their radius periodically, because of the interior mechanisms related with their opacity. The observation of this type of star gives us valuable information about the inner parts of these stars. The increased opacity inside a pulsating star helps to produce heat energy that forces the star to expand. This expansion causes a decrement in the opacity, which results in a shrinkage. The recurrence of this cycle makes the pulsating stars a fascinating candidate for astronomical observation. The observations of the light variation from a pulsating star results in a wave shaped variation of light over time (Fig. 1).

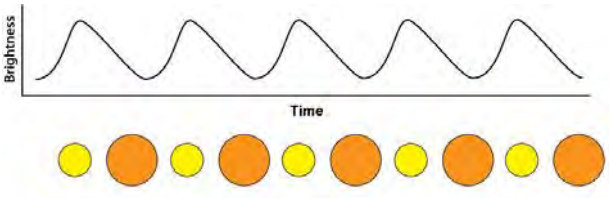


Figure 1. The light curve of a pulsating star in comparison with its radius (Credit: http://www.space-exploratorium.com)

In the case of multiperiodic pulsating stars this wave shape takes a very complicated form and it can only be

decomposed to several sine like variations with certain frequencies (f), amplitudes (A) and phase shifts (φ) by applying frequency analysis. The similarity between observed -wave shaped- light from pulsating stars and a superposed simple sound wave is used in converting the stellar oscillations to audible sounds in our study.

In order to audify the detected oscillation frequencies of a pulsating star, we used the method described by [5]. The author defined three dimensionless parameters based on the pulsation characteristic of a star: the first parameter is Relative Frequency, $f' = f_i/f_{min}$, which is the ratio of a given frequency to the minimum frequency in detected group. The second parameter is the Loudness Parameter, $L = A_i/A_{max}$, which is the ratio of the amplitude value for a given frequency to the maximum amplitude value among the frequency group. The last parameter, $p = \varphi_i/\varphi_{min}$,is the Starting Parameter of the signal. It gives us the difference between the phase shift of a wave and the minimum phase shift value of the group. A light variation profile obtained from the star can be converted to a sound wave by moving the minimum frequency value to a desired frequency in the audible range and by keeping the relation between frequencies, amplitudes and phase shifts. We used five pulsating stars (V465 Per, HD 217860, 16 Lac, BG CVn, KIC 6382916) to produce sounds from the analysis of their observational data. As an example, we give the pulsation parameters (f, A, φ) , related dimensionless parameters (f', L, p) and the result of the multiplication with C_4 for one of our stars, V465 Per Table 1.

f (d^{-1})	A (mmag)	φ	f'	L	p	f'xC_4 (Hz)
14.040	3.5	-0.14	1.023	1.000	0.00	267.647
17.208	2.3	2.05	1.254	0.657	2.19	328.084
33.259	1.7	1.93	2.424	0.488	2.07	634.191
13.721	1.1	3.55	1.000	0.314	3.69	261.630

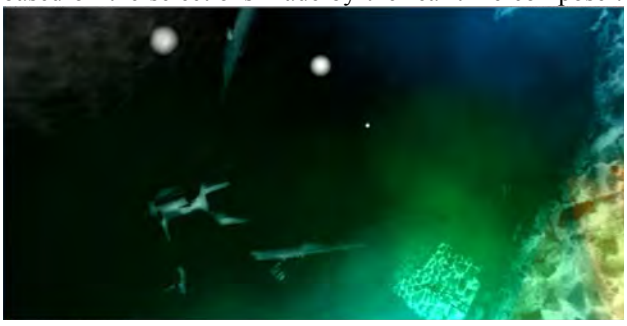
Table 1. The parameters for δ Sct type pulsating star V465 Per. The pulsation parameters (f, A, φ) are taken from [10]. Note that $f_{min} = 13.721$ d^{-1} $A_{max} = 3.5$ mmag and $\varphi_{min} = -0.14$. The frequency value of C_4 , 261.630 Hz, is taken from [11].

For the generation of sound waves from these dimensionless parameters AUDACITY was used. The calculated relative frequencies for a star was multiplied by the frequency value of fourth octave C (see Table 1). The loudness and the starting times are also arranged according to appropriate values. For instance, when converting one observed frequency, say 14.040 d^{-1}, of the star V465 Per to audible range we follow these steps: (i) we multiplied the dimensionless relative frequency by the frequency value of C4, then we entered the new frequency value (i.e. 267.647 Hz) as the frequency of a sound wave. (ii) the Loudness parameter (1.000) was entered directly to the program as the normalized amplitude value. (iii) The starting time parameter (0.00) was set as the starting time of the sound in AUDACITY. Since we have 4 observed frequencies for this star we repeated the process for each of the 4 frequencies listed in Table 1, therefore, we obtained 4 different superposed sound waves characterised by the calculated parameters given in the table. Finally these sound waves were recorded to a digital sound file. We hope to expand on this method with orchestral resynthesis once our system as expanded beyond the early prototyping stages. Below we detail an initial implementation utilizing the audio files we created as described here.

4. OPERA REALISATION

The bulk of the project is realized using the Unity-3d engine and standard assets, with some additions from the Unity app store, most notably the Leap Motion Project Orion Beta, which vastly improves the quality of tracking possible with the Leap Motion camera in comparison the previous assets. The standard character controller included with the Oculus Rift assets was not appropriate due to our intention to port the system to Google Cardboard[2], after the initial development done with the Oculus Rift DK2[3] and Leap Motion[4] camera. Initially there were some problems stemming from the loss of Oculus Rift DK2 Mac OS X support, these had to be overcome by porting the project to a Windows 10 development environment. An important element is the *InstantVR Free* asset from *Passer VR* (http://serrarens.nl/passervr/), which made it possible to port between different VR setups and platforms relatively easy.

Set design was made easier through importing freely available photo-scanned models of historical artifacts or with standard assets as well, saving time on 3d modelling (Fig 4.). The majority of the set is actually a 360-degree photograph taken in one of the “temples”; this was processed into a skybox using the *Panorama to Skybox* asset after being edited into a night-like scene in Photoshop. Some stitching lines are still visible, but they are mostly obscured with particle systems ranging from fog to fire and light. The actual characters in the scene are animated by pre-recorded animations, which are triggered based on the selections made by the real-time composer.



[2] See https://www.google.com/get/cardboard/
[3] See https://www.oculus.com/en-us/dk2/
[4] See https://www.leapmotion.com/

Fig 2. The stage from the real-time composers perspective, 3 stars at different levels of luminosity, in the foreground a fog particle system

5. USER INTERFACE

Fig 3. A user manipulates the interface with the HMD mounted Leap Motion Controller

The real-time composer controls the playback of material by selecting single stars with their hand movements (pinch gesture). Once a particular star is selected, its partials can be used to manipulate the audio of animal calls related to the pictograms featured on reliefs at the Göbeklitepe site. The user is able to manipulate these sound files from within the VR environment through the Leap Motion controller and the Unity audio SDK. Visually the stars themselves increase and decrease in luminosity in accordance with the relative loudness of each group (Fig. 2). For example, the real-time composer orients their hand along the axis of a particular star and their finger positions affect the amplitude of the sine waves related to that star. In the case of V465, the user controls the volumes of 4 sine waves with the degree of extension of their pinky, ring, middle and index fingers (Fig 3). Using gesture recognition the user can also open a HUD, populated by some of the pictograms found throughout the Göbeklitepe site. Selecting one of these pictograms loads a sound file related to the particular animal represented i.e. bison, wild boar, crocodile etc. These sounds can be used with the *SpectraScore~*[5] Max/MSP abstraction to generate spectral music, including scores. Various audio effects allow the user to modify the source sounds in real-time with their movements.

6. CONCLUSION

As this project is in its early stages there is much room for improvement in terms of the interface and software in general. Mainly though, the level of latency experienced between the real-time composers actions and sounding results needs to be improved to create a smoother 'sound-bonding' [10] effect. In order to achieve this, a new system may have to be created relying on playback of samples from the audiences HMDs to reduce network strain. This would hopefully be done with samples of acoustic instruments such as is currently done with *SpectraScore* via MIDI or OSC. Spatialisation would then become a further layer of complexity, due to the strain of performing DSP in a smartphone headset.

All in all our success at bringing together techniques of spectral music composition methods and starlight audification points at the relative ease through which new algorithms can be imported into existing algorithmic music composition frameworks. Since this project was realised in VR, the exploratory nature of real-time composition was brought into focus through the use of 'source objects', that is, 'material objects'(Culverwell[6]) that have been analysed and re-represented in a musical form as spectral morphemes (representing the physical forms from which they were derived). This referential expressionist form of Spectralism creates new possibilities for a kind of figurative interaction between 'Gestalten' that are otherwise incomparable. Thanks also to an extensive array of virtualised real-world objects available in online collections and stores (i.e Sketchfab, Turbosquid, Unity Asset Store), and the ever increasing documentation surround the mapping of the sky above the Earth, the possibilities for sonification with the techniques described here will continue to grow and increase in relevance for proponents of the *Gesamtkunstwerk*.

[5] See https://github.com/benedictcarey/SpectraScore-beta-0.4

[6] http://www.oxforddictionaries.com/definition/english/material-object

Fig 4. Set design detail including 360 degree photograph to set conversion, front and centre is one of the pictograms used to trigger animal calls.

7. REFERENCES

[1] Zayed, Ahmed I. *Handbook of function and generalized function transformations*. CRC press, 1996. Pp. 582

[2] Fineberg, J. N. (1999). 'Sculpting sound: an introduction to the spectral movement - its ideas, techniques and music'. PhD thesis, Columbia University.

[3] Féron, François-Xavier. "Gérard Grisey: première section de Partiels (1975)."*Genesis. Manuscrits–Recherche–Invention* 31 (2010): 77-97.

[4] The FFT - An algorithm the whole family can use, *Computing in Science & Engineering,* January/February 2000, **2**, Number 1, pp. 60--64.

[5] Ulas, B., 2015, arXiv:1507.0730

[6] Teodorescu-Ciocanea, L. (2003). 'Timbre versus spectralism'. *Contemporary Music Review*, *22*(1-2), 87-104.

[7] Malherbe, C., Fineberg, J., & Hayward, B. (2000). 'Seeing Light as Color; Hearing Sound as Timbre'. *Contemporary Music Review*, *19*(3), 15-27.

[8] Moscovich, Viviana. "French spectral music: An introduction." *Tempo* (1997): 21-27.

[9] Meyer, L. B. (2008). Emotion and meaning in music, University of Chicago Press.

[10] Kim, S.-L. and Lee, S.-W., 1998, Astronomy and Astrophysics Supplement, v.128, p.111-116

[11] Suits, B. H., 1998, http://www.phy.mtu.edu/~suits/notefreqs.html

[10] Smalley, D. (1997). "Spectromorphology: explaining sound-shapes." Organised sound 2(02): 107-126.

[11] Bundler, David. "Interview with Gerard Grisey." 20th Century Music (1996).

Rethinking the audio workstation: tree-based sequencing with i-score and the LibAudioStream

Jean-Michaël Celerier
LaBRI, Blue Yeti
Univ. Bordeaux, LaBRI, UMR 5800,
F-33400 Talence, France.
Blue Yeti, F-17110 France.
jcelerie@labri.fr

Myriam Desainte-Catherine
LaBRI, CNRS
Univ. Bordeaux, LaBRI, UMR 5800,
F-33400 Talence, France.
CNRS, LaBRI, UMR 5800,
F-33400 Talence, France.
INRIA, F-33400 Talence, France.
myriam@labri.fr

Jean-Michel Couturier
Blue Yeti, F-17110 France.
jmc@blueyeti.fr

ABSTRACT

The field of digital music authoring provides a wealth of creative environments in which music can be created and authored: patchers, programming languages, and multi-track sequencers. By combining the I-SCORE interactive sequencer to the LIBAUDIOSTREAM audio engine, a new music software able to represent and play rich interactive audio sequences is introduced. We present new stream expressions compatible with the LIBAUDIOSTREAM, and use them to create an interactive audio graph: hierarchical stream and send - return streams. This allows to create branching and arbitrarily nested musical scores, in an OSC-centric environment. Three examples of interactive musical scores are presented: the recreation of a traditional multi-track sequencer, an interactive musical score, and a temporal effect graph.

1. INTRODUCTION

Software audio sequencers are generally considered to be digital versions of the traditional tools that are used in a recording studio: tape recorders, mixing desks, effect racks, etc.

Most of the existing software follow this paradigm very closely, with concepts of tracks, buses, linear time, which are a skeuomorphic reinterpretation of the multi-track tape recorder [1]. On the other side of the music creation spectrum, we find entirely interaction-oriented tools, such as MAX/MSP (Cycling 74'), PUREDATA, CSOUND, or SUPERCOLLIDER. They allow the user to create musical works in programming-oriented environments. In-between are tools with limited interaction capabilities but full-fledged audio sequencing support, such as Ableton LIVE, or Bitwig STUDIO. The interaction lies in the triggering of loops and the ability to change the speed on the fly but is mostly separate from the "traditional" sequencers integrated in these software packages.

In this paper, we present a graphical and hierarchical approach to interactive audio sequencing. We integrate the LIBAUDIOSTREAM audio engine in the interactive control sequencer I-SCORE. This takes form as an audio sequencing software package that allows the user to author music on a time-line with the possibility to arbitrarily nest sounds and effects and trigger sounds interactively while keeping the logical coherency specified by the composer. An extension is introduced to arrange audio effects in an interactive temporal graph. For instance, instead of simply applying a chain of effects to an audio track, it is possible to apply temporal sequences of effects: an effect would be enabled for ten seconds, then, if an external condition becomes true, another effect would be applied until the musician chooses to stop it.

We will first present the existing works in advanced audio sequencing and workstations, and give a brief presentation of both I-SCORE and the LIBAUDIOSTREAM. Then, the new objects introduced in order to integrate these software packages together, allowing for rich audio routing capabilities, will be explained. Finally, three examples of usage in the graphical interface of I-SCORE will be provided: a recreation of a standard multi-track player, an interactive score, and an effect graph applied to a sound.

2. EXISTING WORKS

Outside of the traditional audio sequencer realm, there are multiple occurrences of graphical environments aiming to provide some level of interactivity.

Möllenkamp presents in [2] the common paradigms used when creating music on a computer: score-based with MUSIC and CSOUND [3], patch-based with MAX/MSP (Cycling 74') or PUREDATA [4], programming-based with SUPERCOLLIDER [5] and many of the other music-oriented programming languages, trackers such as FASTTRACKER which were used to program the music in early console-based video games, and multitrack-like such as Steinberg CUBASE, Avid PRO TOOLS. Ableton LIVE and Bitwig STUDIO are given their own category thanks to the ability to compose clips of sound interactively.

DRILE [6] is a virtual reality music creation software package. Loops are manipulated and bound together in a 3D environment, through instrumental interaction. Hierarchy

is achieved by representing the loops in a tree structure.

KYMA [7] is a hybrid software and hardware environment for sound composition. It offers multiple pre-made facilities for sound creation such as multi-dimensional preset interpolation, sound composition by addition and mutation, or sequential and parallel sound composition on a time-line.

AUDIOMULCH [8] is an environment for live music performance, which also provides preset space exploration thanks to the Metasurface concept. Cantabile PERFORMER [1] is also an environment geared towards live performance, with the ability to trigger sounds, and a temporal ordering. It is closer to the cue metaphor than the sequencer metaphor.

Mobile and web applications are being increasingly used to create music, but their are often embedded in a bigger score or framework and act more as an instrument than in other systems. An interesting example of a web-based sequencer is JAMON [9] which allows multiple users to collaboratively and interactively author music by drawing within a web page interface. A deeper overview of the collaborative music authoring environments is given in [10].

Finally, modern video game music engines such as FMOD and AudioKinetic WWISE allow some level of interactivity, i.e when an event occurs in a video game, a sound will be played. Automation of parameters is possible, and these environments are geared towards three-dimensional positioning of sound and sound effects such as reverb, echo.

For low-level audio engines, one of the predominant methods is the audiograph. Prime examples are Jamoma AUDIOGRAPH [11] and INTEGRA FRAMEWORK [12]. Audio processing is thought of as a graph of audio nodes, where the output of a node can go to the input of one or multiple other nodes. Audio workstations such as Magix SAMPLITUDE (with the flexible plug-in routing) and Apple LOGIC PRO (with the Environment) provide access to the underlying audio graph.

3. CONTEXT

In this section, we will present the two tools that we used to achieve audio sequencing: I-SCORE and the LIBAUDIOSTREAM. I-SCORE is an interactive sequencer for parameters, which allows one to position events in time, and introduce interaction points and conditions throughout the score. The detailed execution semantics are given in [13].

The LIBAUDIOSTREAM [14] provides the functionality allowing the authoring of audio expressions through creation and combination of streams. The notion of symbolic date, introduced in an extension of the library, allows the user to start and stop the execution of streams at a time and date not known until the performance.

The goal of this work is to bind the audio capabilities of the LIBAUDIOSTREAM with the I-SCORE execution engine and graphical interface, in order to allow the creation of rich hierarchic and interactive musical pieces.

[1] https://www.cantabilesoftware.com/

3.1 Presentation of i-score

The original goal of I-SCORE is to communicate and coordinate other software in a synchronized manner, through the OSC protocol. The software can be used to send automations, cue-like OSC messages at a given point in time, and call arbitrary JavaScript functions, in a sequenced environment. It supports arbitrary nesting: a score can be embedded in another recursively. This is similar to the notion of group tracks in many other sequencers, but without a depth limit. Besides, there is no notion of "track" per se; rather, the composer works with temporal intervals which contain arbitrary data that can be provided by plug-ins.

Multiple methods of interactivity are available in I-SCORE: trigger points, conditions, mappings, speed control.

- Interactive triggers are used to block and synchronize the score until a specific event happens. For instance, when an OSC parameter fulfills a condition, such as /a/b ≤ 3.14, then a part of the score can continue.
- Conditions enable the execution or disabling of part of the score according to a boolean condition. This makes if-then-else or switch-case programming constructs easy to implement in a temporal way.
- Mappings allow the user to map an input parameter to an output parameter, with a transfer function applied to the input.
- The execution speed of hierarchical elements can be controlled during the execution.

A span of time in I-SCORE might have a fixed or indefinite duration; we refer to this span as a Time Constraint (TC) since it imposes both a logical and temporal order to the elements before and after it.

A TC may contain data in the form of processes: automations, mappings, but also loops and scenarios; a scenario is the process that allows nesting. When the TC stops, all its processes are killed recursively.

TCs are linked together with Time Nodes, which allow for synchronization and branching of multiple streams of time.

An example of the temporal syntax of I-SCORE is presented in fig. 1. It is for instance used by Arias and Dubnov in [15] to construct a musical environment adapted to improvisation by segmenting pre-recorded audio phrases, to allow constrained improvisation according to high-level musical structures. The resulting generated structures bear similarity with the session concept in Ableton LIVE: one can loop dynamically over particular sections of a sound file.

3.2 Presentation of the LibAudioStream

The LIBAUDIOSTREAM [14], developed at GRAME, is a C++ library allowing the user to recreate the constructs commonly found in multi-track sequencers directly from code; it also handles communication with the sound card hardware via the common audio APIs found on desktop operating systems.

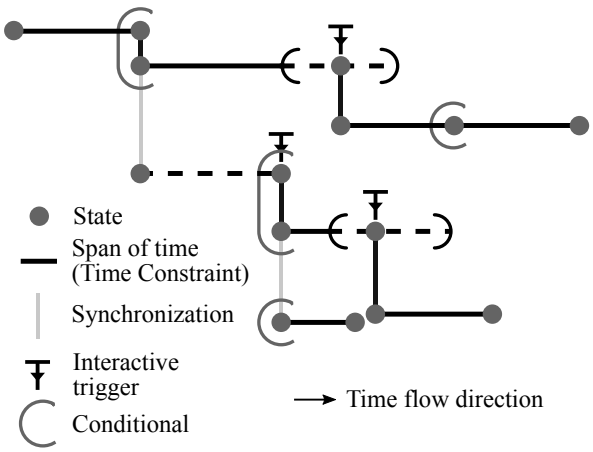


Figure 1. Part of an I-SCORE scenario, showcasing the temporal syntax used. A full horizontal line means that the time must not be interrupted, while a dashed horizontal line means that the time of this Constraint can be interrupted to continue to the next part of the score according to an external event

One of the advantages of using a library instead of a graphical interface is that it provides scripting capabilities to the composer and makes algorithmic music composition easier. It has been used with success in OPENMUSIC [16].

Audio sounds and transformations are modeled via streams. The following operations are applied to these streams: serial and parallel composition, mixing, and multi-channel operations. Streams are bound together in order to construct complex audio expressions. For instance, two sound files can be mixed together with a Mix stream expression:

```
auto sound = MakeMixSound(
   MakeReadSound("a.wav"),
   MakeReadSound("b.wav"),
   0.75);
```

A stream can then be played via an audio player, with audio sample accuracy:

```
StartSound(audioplayer, sound, date);
```

The play date must not necessarily be known in advance thanks to the notion of symbolic date [2] . Finally, FAUST [17] audio effects can be applied to the streams.

4. PROPOSED AUDIO SYSTEM

In this section, we will explain the audio routing features offered by the software.

First, we introduce new audio streams that allow a LIBAUDIOSTREAM expression to encapsulate the execution of a virtual audio player, in order to allow for hierarchy.

We make the choice to allow for hierarchy by mixing the played streams together, in order to follow the principle of least astonishment [18]: in most audio software, the notion of grouping implies that the grouped sounds will be mixed together and routed to a single audio bus.

[2] Symbolic dates in the LibAudioStream are dates that can be set dynamically at run time.

Then, we present the concept of audio buses integrated to the LIBAUDIOSTREAM, with two special Send and Return streams.

Finally, we exhibit the translation of I-SCORE structures in LIBAUDIOSTREAM expressions, which required the creation of a dependency graph between audio nodes.

4.1 Group audio stream

In order to be able to apply hierarchical effects on the streams, and handle interactivity in recursive groups, we have to introduce a way to use sound hierarchy in the LIBAUDIOSTREAM.

Our method employs two elements:

- A particular audio player that will be able to sequence the starting and stopping of interactive sounds. Such players already exist in the LIBAUDIOSTREAM but are tailored for direct output to the sound card
- A way to reintroduce the player into the stream system, so that it is able to be pulled at regular intervals like it would be by a hardware sound card while being mixed or modified by subsequent stream operators.

We introduce matching objects in the LIBAUDIOSTREAM:

- A Group player. This is a player whose processing function has to be called manually. Timekeeping supposes that it will be pulled in accordance with the clock rate and sample rate of the sound card.
- A Group audiostream. This particular audiostream, of infinite length, allows the introduction of a Group player in a series of chained streams and takes care of having the Player process its buffers regularly.
- A finite loop audiostream. This stream loops over its content after a given duration.

The execution time of the nested objects will be relative to the start time of the Group audiostream.

4.2 Send and return audio streams

In order to be able to create temporal effect graphs, we introduced another couple of objects.

The Send audiostream by itself is a pass-through: it just pulls the stream it is applied to. It possesses the same definition: same length, same number of channels. The Return audiostream, constructed with a Send stream, makes a copy of the data in the Send stream and allows it to be used by the processing chain it is part of. For instance, this means that a single sound source can be sent to two effect chains in parallel.

The Return stream is unlimited in length: to allow for long-lasting audio effects like reverb queues or delays, we suppose that we can pull the data from the Send stream at any point in time. If a Return stream tries to fetch the data of a Send stream that has not started yet, or that has already finished, a buffer of silence is provided instead.

The Send stream must itself be pulled regularly by being played as a sound, either directly or by a stream that would encapsulate it.

An example of such an audiostream graph is presented in fig. 2.

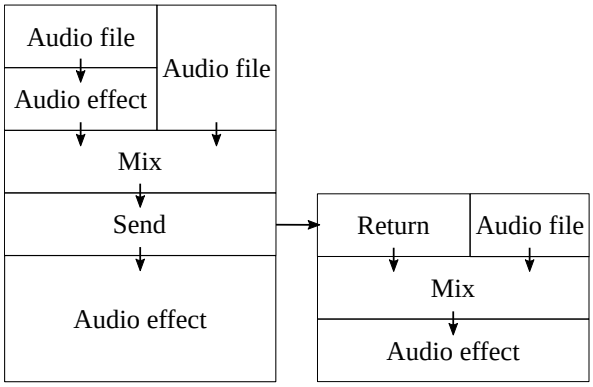


Figure 2. An example of audio stream composition with the Send and Return objects. An arrow from A to B means that B pulls the audio data from A.

4.3 Audio processes

We provide multiple audio processes in I-SCORE, that map to the existing LIBAUDIOSTREAM structures.

- Effect chain process: register multiple audio effects one after the other. For instance:
  Equalizer → *Distortion* → *Reverb*.
  Currently only FAUST effects or instruments are supported. Interfaces are provided to allow the extension to other audio plug-in formats.
- Input process: allows the introduction of the audio input of a sound card to the stream.
- Sound file: reads a sound file from the file system.
- Explicit send and return processes for manual routing.
- Mixing process: it exposes a matrix which allows to adjust the percentage of each sound-generating process going to each input process, send, and parent.

An important feature of audio workstations is the support for automation, that is, controlling the value of a parameter over time, generally with piecewise continuous functions. In I-SCORE, automation is achieved by sending OSC messages to a remote software package. The OSC messages tree is modeled as an object tree. We present the loaded effect plug-ins to this object tree, so that automations and mappings can control audio effects and audio routing volume.

A screen capture of a TC with processes is given in fig. 3.

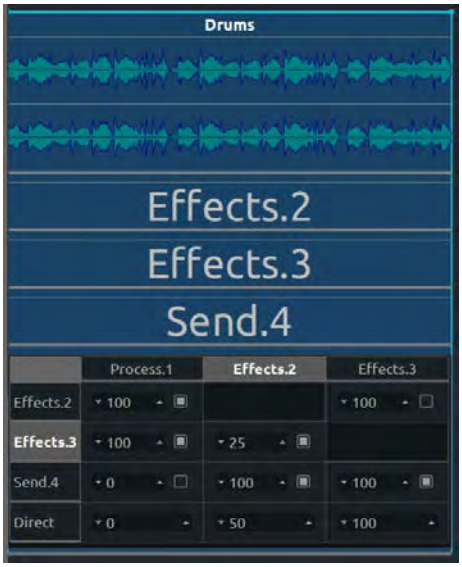


Figure 3. An example of a TC loaded with audio processes in I-SCORE. Selecting a particular process shows a complete widget for editing the relevant parameters. On a single TC, there can be only a single Mixing process (the table at the bottom), but there is no limit to the amount of other processes i.e there can be multiple sound files, etc.

4.4 Stream graph

One problem caused by the presence of routing is that it is possible to create a data loop: if a Send is directly or indirectly fed its own data through a Return, the output would be garbage data. The Return would be asked to read the currently requested output from the Send which has not been written yet.

To prevent this, we create a graph where:

- Vertices are the sound generating elements associated with their output Send, for example e.g. an audio file reader, hierarchical elements, etc.
- Edges are the connections going from a send to a return, or from an element to the element it is mixed in.

The graph, implemented with the Boost Graph Library [19] can then be used to check for acyclicity. The user will be notified if that is not the case.

We provide here the method to build the graph.

Vertices are created recursively from the TCs in I-SCORE: an I-SCORE document is entirely contained in a top-level TC.

First, we iterate through all the processes of the given constraint. If the process is hierarchical (Scenario, Loop), then we call the algorithm recursively on the process.

In the case of the Scenario, it means that we call recursively on all its Constraints. In the case of the Loop, we call recursively on its loop pattern Constraint. In both cases, we create a Group vertex to model the process. Edges are to be added from each stream in the hierarchical time-line, to the group stream.

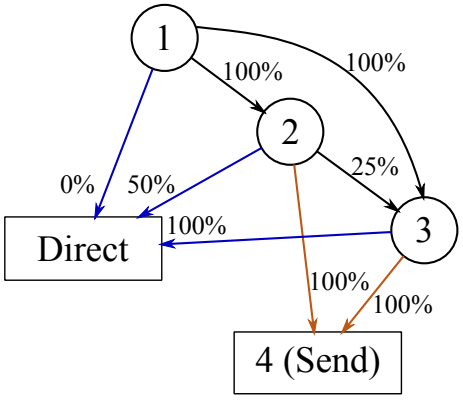


Figure 4. Translation of the TC of fig. 3 in a dependency graph. The edges in black represent the intra-Constraint connections. The edges in blue (resp. orange) represent a connection to a visible output of the Constraint. The percentages represent the level of mixing of the stream. *Direct* corresponds to the signal that will be sent at the upper level of hierarchy.

If the process is a send or a return, we create a corresponding vertex. Then, we create inner Sends for all the streams and a vertex for the Constraint itself.

Once all the vertices are created, the edges are added as mentioned before.

As mentioned before, there is an ordering between nodes of the graph: the parent-most vertex has to be pulled before the others to reflect the causality.

Inside a TC, causality also has to be enforced. Since a mixing matrix is provided, we have to ensure that an effect bus cannot be routed back into itself creating a loop. To prevent this at the user interface level, we disable by default the mixing of audio effect chains into each other. In fig. 4, we show the resulting graph for a TC.

When the graph is verified acyclic, we perform the stream creation by iterating over the list of topologically sorted vertices.

4.5 Stream creation

In this section we will detail the stream creation for particular elements.

4.5.1 Scenario

An I-SCORE scenario is an arrangement of temporal structures, as shown in fig. 1; it is an independent time-line. Since the execution and duration of these structures can change at run-time due to interactivity and hierarchy, it is not meaningful to directly use the tools provided by the LIBAUDIOSTREAM: sequence stream, parallel stream, mix stream. We instead use the Group player to organize our elements in time.

The creation of the Scenario stream is done as follows:

1. A Group player is created.

2. For each Time Node in the scenario, a symbolic date is generated.

3. For each TC in the scenario, a stream is built; it is started and stopped at the symbolic date matching its start and end Time Nodes in the group player.

The Audio stream of this process is the group player. In order to enforce sample-accuracy whenever possible, if the i-score structures have a fixed date, we preset this date to its computed value. If there is no interactivity involved, a sound directly following on from another will begin to play from the audio sample past the end of the first one. As soon as a sound's execution time is fixed, an algorithm checks for all the following sounds whose date could also be fixed.

4.5.2 Loop

Due to their interactive nature, loops in I-SCORE can be entirely different from one iteration to another. They are more similar to imperative programming `do-while` constructs, than audio sequencer loops. This prevents us from directly using the LIBAUDIOSTREAM's loop stream, since it expects a looping sound of finite duration. Instead, if the loop is interactive, we wrap the loop pattern TC's audiostream in a Group player, reset the stream and start it again upon looping. If the loop is not interactive, we can reset it at a fixed interval of time with the fixed loop stream introduced earlier. This allows for sample accurate hierarchical looping with I-SCORE's process semantics.

4.5.3 Time Constraint

As explained earlier, a TC is a process container. Such processes can be the sound processes presented in section 4.3, and the control processes such as automation, etc.

The creation of the Constraint audio stream is done as follows:

1. For each sound-generating process, a stream and a send are created.

2. For each effect chain, the effects are instantiated and an effect stream is created with a mix of the returns of the elements to which this effect applies. A send is also created.

3. The mixing matrix is used to create mix audio streams from the sends and returns, which are routed either in the user-created sends, or in the stream corresponding to the TC. A time-stretching audio stream is inserted before the send: it is linked to the execution speed of the TC in I-SCORE which can vary interactively.

4.5.4 A note on real-time performance

Since a real-time audio input is provided, we ought to be able to use the system as a multi-effect, so as to introduce the lowest possible latency. The time-stretching effect itself may impose a latency high enough to make playback through the system impossible. Similarly, Sends and Returns must operate at the same playback speed; else, the Return will either fetch not enough data, or skip data from the Send it is listening to.

To solve this, when creating the graph, the parents of each Input, Send, and Return nodes are recursively marked with a flag to indicate real-time processing. The TCs with this flag will not be able to be time-stretched, and will only be affected by the latency due to the effects manually introduced by the composer.

5. EXAMPLES

We present in this part three examples of usage of the presented system.

5.1 Recreation of a multi-track sequencer

The first example, in fig.5, is a recreation of the multi-track audio sequencer metaphor, with the primitives presented in this paper.

This score has three tracks, **Guitar**, **Bass**, and **Drums**, which are implemented with three TCs. Each TC has a Sound process and an Effect process; the Mixing process is hidden for clarity. The bass track is a looping one-note sound. Automations are applied either at the "track" level, as for the drums, or at the "clip" level, as for the guitar **outro** clip. However, in the model there is no actual difference between track and clip, it is solely a particular organization of the score.

5.2 Interactive scenario

The second example, in fig.6, gives an overview of the interactive possibilities when creating a song.

The score behaves as follows: for a few seconds, **intro** will play. Then, if an external event happens, like a foot switch being pressed, multiple things may happen:

- In all cases, the **eqcontrol** part will play, and automate a value of a global effect.
- If a first condition is true (**case1**), then **case1.B** will start playing immediately, and **case1.A** will start playing after a slight delay. If another external event happens, **case1.A** will stop playing immediately.
- If a second condition is true, at the same time, **case2** will start playing.
- After **eqcontrol** finishes, a hierarchical scenario **outro** is played, which contains two sounds and a parameter mapping.

If no external event happens, after some time, when reaching the end of the triggerable zone delimited by the dashed line, the triggering occurs regardless of user input.

5.3 Temporal effect graph

This last example, in fig. 7 shows how to arrange not sound, but sound processing temporally. In this case, we have a sound playing, which is routed in the send process. Then, a hierarchical scenario with multiple TCs is used. Two TCs have return processes connected to the previously created send. Automations are applied to parameters of these effects.

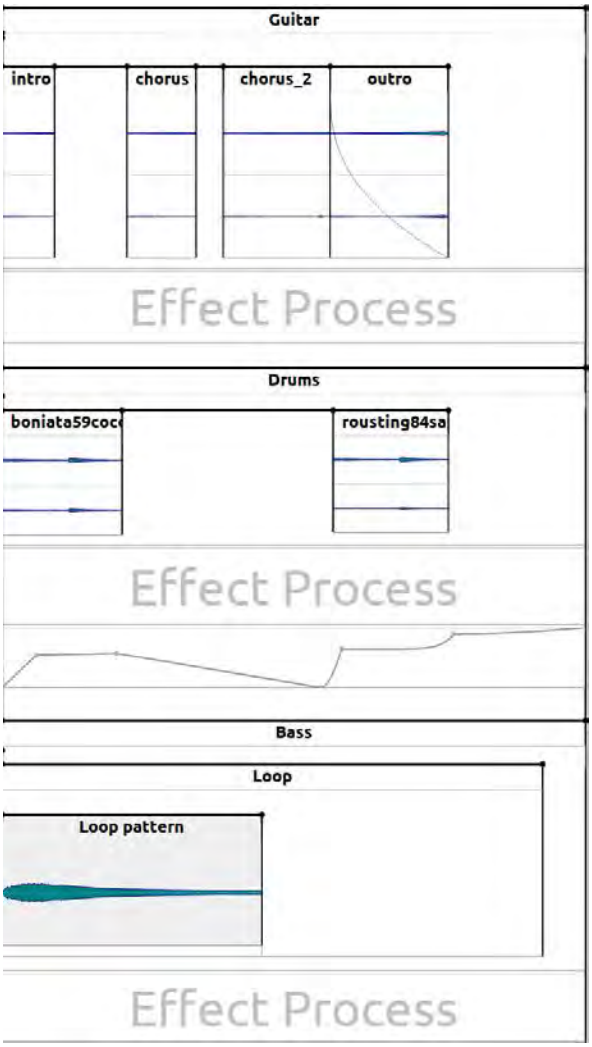


Figure 5. Multi-track sequencing.

Second effect will be triggered after an external event happens. By using loops, effects, and TCs with infinite durations, this same mechanism would allow one to simulate a guitar pedal board with switchable effects, and to create temporal transitions between the various sounds.

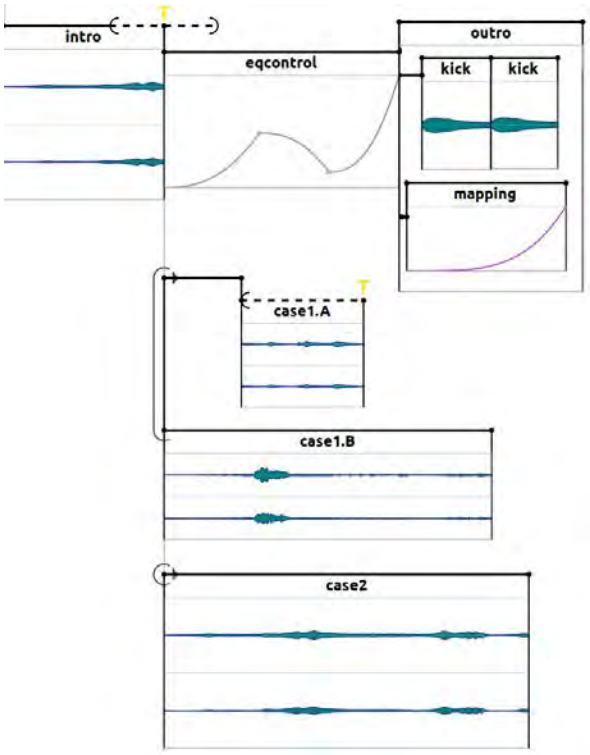


Figure 6. An interactive musical score.

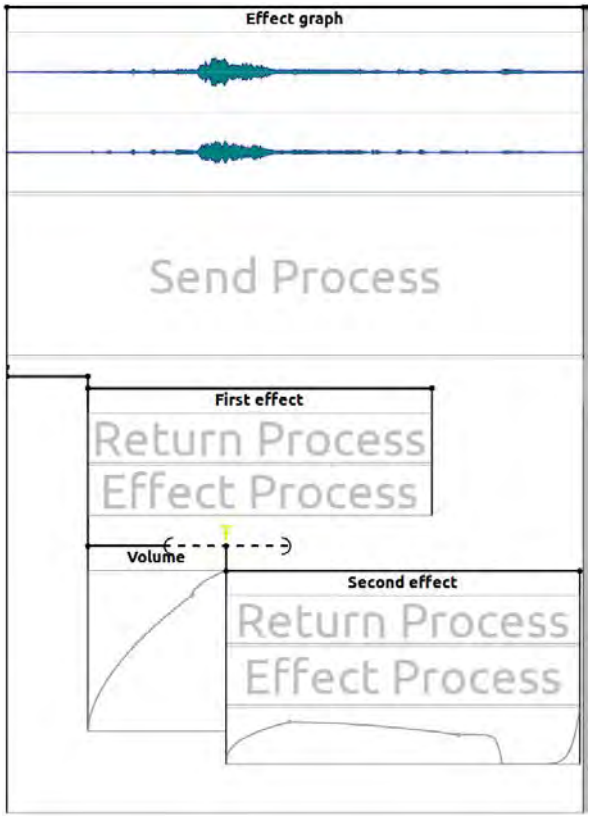


Figure 7. Temporal effect graph applied to a sound.

6. CONCLUSION

We presented a computer system for creating interactive music, which extends the audio sequencer metaphor. New kind of streams enabling hierarchy and audiograph-like behavior are introduced to the LIBAUDIOSTREAM, which is then binded to the I-SCORE primitives for specifying and scoring time and interaction. Three examples present the various musical possibilities that are offered through this system.

However, there are currently some key differences to more traditional musical environments: for one, musical notation and concepts are absent from the system. All durations are expressed in seconds or milliseconds, instead of beats or any subdivision as they would in other environments. A possible extension to the I-SCORE execution engine would be to take into account beats for triggering, which would allow the user to synchronize multiple hierarchical loops to a beat and may be useful for some genres of music, such as electronica or rock.

Likewise, the system mostly handles audio and OSC data; MIDI is implemented at a primitive level. Another unhandled question is effect delay compensation: sometimes, audio algorithms must introduce multiple frames of latency in their processing chain, for instance because they have to accumulate a certain amount of data. This is not taken into account here, hence seemingly synchronized sounds may desynchronize themselves if this latency is not accounted for.

Finally, in many cases optimizations could be achieved to reduce the amount of data structures being created. For instance, when a single sound file is in a TC, a simpler stream expression could be created.

The next steps for this research includes these points, work on sound spatialization, and interactive edition: modifying the score while it is already playing.

Acknowledgments

This research was supported by the SCRIME (Studio de Création et de Recherche en Informatique et Musiques Expérimentales, `scrime.labri.fr`) which is funded by the French Culture Ministry. SCRIME is a GIS (Group of Interest in Science and Art) with University of Bordeaux, Bordeaux INP, Bordeaux City, CNRS (National Scientific Research Center), Région Nouvelle Aquitaine (Aquitaine Regional Council) and the DRAC (Regional Direction of Culture). This work was also supported by an ANRT CIFRE convention with the company Blue Yeti under funding 1181-2014. The authors wishes to thanks Stéphane Letz for his help with the LibAudioStream.

7. REFERENCES

[1] A. Bell, E. Hein, and J. Ratcliffe, "Beyond Skeuomorphism: The Evolution of Music Production Software User Interface Metaphors," *Journal on the Art of Record Production*, 2015.

[2] A. Möllenkamp, "Paradigms of Music Software Development," in *Proceedings of the 9th Conference on Interdisciplinary Musicology*, 2014.

[3] B. Vercoe and D. Ellis, "Real-time csound: Software synthesis with sensing and control," in *Proceedings of the International Computer Music Conference*, 1990, pp. 209–211.

[4] M. Puckette *et al.*, "Pure data: another integrated computer music environment," *Proceedings of the second intercollege computer music concerts*, pp. 37–41, 1996.

[5] J. McCartney, "Rethinking the computer music language: Supercollider," *Computer Music Journal*, vol. 26, no. 4, pp. 61–68, 2002.

[6] F. Berthaut, M. Desainte-Catherine, and M. Hachet, "Drile: an immersive environment for hierarchical live-looping," in *New Interfaces for Musical Expression*, 2010, p. 192.

[7] C. Scaletti, "The kyma/platypus computer music workstation," *Computer Music Journal*, vol. 13, no. 2, pp. 23–38, 1989.

[8] R. Bencina, "The metasurface: applying natural neighbour interpolation to two-to-many mapping," in *New Interfaces for Musical Expression*, 2005, pp. 101–104.

[9] U. Rosselet and A. Renaud, "Jam On: a new interface for web-based collective music performance," in *New Interfaces for Musical Expression*, 2013, pp. 394–399.

[10] R. Fencott and N. Bryan-Kinns, "Computer musicking: HCI, CSCW and collaborative digital musical interaction," in *Music and Human-Computer Interaction*. Springer, 2013, pp. 189–205.

[11] T. Place, T. Lossius, and N. Peters, "The jamoma audio graph layer," in *Proceedings of the 13th International Conference on Digital Audio Effects*, 2010, pp. 69–76.

[12] J. Bullock and H. Frisk, "The integra framework for rapid modular audio application development," in *Proceedings of the International Computer Music Conference*, 2011.

[13] J.-M. Celerier, P. Baltazar, C. Bossut, N. Vuaille, J.-M. Couturier, and M. Desainte-Catherine, "OSSIA : Towards a Unified Interface for Scoring time and Interaction," in *Proceedings of the TENOR 2015 Conference*.

[14] S. Letz, "Spécification de l'extension LibAudioStream," Tech. Rep., Mar. 2014. [Online]. Available: https://hal.archives-ouvertes.fr/hal-00965269

[15] J. Arias, M. Desainte-Catherine, and S. Dubnov, "Automatic Construction of Interactive Machine Improvisation Scenarios from Audio Recordings," in *4th International Workshop on Musical Metacreation (MUME 2016)*, Paris, France, 2016.

[16] D. Bouche, J. Bresson, and S. Letz, "Programmation and Control of Faust Sound Processing in OpenMusic," in *Joint International Computer Music/Sound and Music Computing Conferences*, 2014.

[17] Y. Orlarey, D. Fober, and S. Letz, "Faust: an efficient functional approach to DSP programming," *New Computational Paradigms for Computer Music*, vol. 290, 2009.

[18] P. Seebach, "The cranky user: The Principle of Least Astonishment," in *IBM DeveloperWorks*, 2001.

[19] J. G. Siek, L.-Q. Lee, and A. Lumsdaine, *The Boost Graph Library: User Guide and Reference Manual, Portable Documents*. Pearson Education, 2001.

USING MULTIDIMENSIONAL SEQUENCES FOR IMPROVISATION IN THE OMAX PARADIGM

Ken Déguernel
Inria
STMS Lab, Ircam / CNRS / UPMC
ken.deguernel@inria.fr

Emmanuel Vincent
Inria
emmanuel.vincent@inria.fr

Gérard Assayag
STMS Lab, Ircam / CNRS / UPMC
gerard.assayag@ircam.fr

ABSTRACT

Automatic music improvisation systems based on the OMax paradigm use training over a one-dimensional sequence to generate original improvisations. Different systems use different heuristics to guide the improvisation but none of these benefits from training over a multidimensional sequence. We propose a system creating improvisation in a closer way to a human improviser where the intuition of a context is enriched with knowledge. This system combines a probabilistic model taking into account the multidimensional aspect of music trained on a corpus, with a factor oracle. The probabilistic model is constructed by interpolating sub-models and represents the knowledge of the system, while the factor oracle (structure used in OMax) represents the context. The results show the potential of such a system to perform better navigation in the factor oracle, guided by the knowledge on several dimensions.

1. INTRODUCTION

Current automatic music improvisation systems such as OMax [1] are able to learn the style of a one-dimensional musical sequence (a melody represented by a sequence of pitches or timbral audio features) in order to generate original improvisations by recombining the musical material. This style modeling can be performed live from a musician's playing or offline with a corpus. Several systems have been developed over the years using statistical sequence modeling [2], Markovian models [3] and other machine learning techniques [4]. However, most of these systems do not take the correlations between several musical dimensions (pitch, harmony, rhythm, dynamic, timbre...) into account.

Taking into consideration multiple dimensions and the relations between them has been an issue for systems out of the OMax paradigm. ImproteK [5, 6] makes use of a prior knowledge of a scenario (for example a chord chart) to guide the improvisation. SoMax [7] uses an active listening procedure enabling the system to react to its environment by activating places in its memory. PyOracle [8] uses information dynamics on audio features to create improvisations. Donze et al. [9] use an automaton in order to control the melodic improvisation with information about other dimensions. But in all of these, the actual training is still done on a one-dimensional sequence.

Training on multidimensional sequences has been studied by Conklin et al. [10] with multiple viewpoint systems where different attributes of a melody (such as pitches, intervals, contour...) are linked together for melody prediction on Bach chorales. These systems have also been studied for four part harmonisation [11]. Raczyński et al. use interpolated probabilistic models to do melody harmonisation [12]. This work proposes a flexible way to create a global model from chosen sub-models whose weight can be optimised and can be used in practice since the size of the model is reduced in order to learn the dependencies between dimensions. This method also uses smoothing techniques [13] to reduce overfitting issues that would otherwise arise. Some multidimensional models based on deep neural networks have also been proposed for the harmonisation problem [14] or to create jazz melodies [15]. In this case, the dependencies between dimensions are implicitly represented in the hidden layers.

In this article we present a way to use interpolated probabilistic models to create improvisations taking into account multiple musical dimensions and the correlations between them while keeping the benefits of the OMax paradigm and its factor oracle based representation [16], in particular its linear time oriented graph structure and optimised navigation scheme that make it a proficient tool for improvised performance and interaction. These are well-established methods that can profit from advanced smoothing and optimisation techniques. Moreover, they provide more explanatory models than neural network and therefore can provide us a deeper insight into the studied musical style or the improviser's mind. We combine these models with the factor oracle [17] structure used in OMax, thus creating a new system with a musical training, able to use prior multidimensional knowledge to guide itself in an improvisation context described by the factor oracle.

In section 2, we explain how interpolation of probabilistic models can be used to take multiple dimensions into account for melody generation. Then, in section 3, we introduce a system combining probabilistic models with the

factor oracle. And finally, in section 4 we present some results of experimentations done with this new system.

2. INTERPOLATION OF PROBABILISTIC MODELS

2.1 Method

Our system relies on the work of Raczyński et al. in [12] on automatic harmonisation. We want to create a probabilistic model able to predict the melody given information from different musical dimensions. Let us denote by M_t the melody played at time t, represented by the pitch. We want to predict :

$$P(M_t|X_{1:t}) \tag{1}$$

where $X_{1:t}$ is a set of musical variables from times 1 to t. This model is able to take into account multiple musical dimensions since the musical variables included in $X_{1:t}$ can be from several dimensions.
However, the combinatorics behind such a model are too high, the set of possibilities being the cartesian product of the set of possibilites of each dimension. Therefore such a prediction cannot be used in practice. To make it applicable, we approximate this global model by interpolating several sub-models P_i, which are easier to compute, depending only a subset of the musical variables $A_{i,t} \subset X_{1:t}$. For instance, we can use an n-gram model over a single dimension, or models representing the direct interaction between dimensions, for example, "which note should I play at time t knowing the harmony at this time?".
The interpolation can be linear [18] :

$$P(M_t|X_{1:t}) = \sum_{i=1}^{I} \lambda_i P_i(M_t|A_{i,t}) \tag{2}$$

where I is the number of sub-models and $\lambda_i \geq 0$ are the interpolation coefficients such that

$$\sum_{i=1}^{I} \lambda_i = 1$$

The interpolation can also be log-linear [19] :

$$P(M_t|X_{1:t}) = Z^{-1} \prod_{i=1}^{I} P_i(M_t|A_{i,t})^{\gamma_i} \tag{3}$$

where $\gamma_i \geq 0$ are the interpolation coefficients and Z is a normalising factor :

$$Z = \sum_{M_t} \prod_{i=1}^{I} P_i(M_t|A_{i,t})^{\gamma_i}. \tag{4}$$

The optimisation over the interpolation coefficients enable the system to accept as many sub-models as possible. The most relevant sub-models will have a high interpolation coefficient while irrelevant sub-models will receive an interpolation coefficient close to zero. This could be extended with some sub-model selection similar to Model M [20].

Two methods of smoothing techniques are used, the latter being a generalisation of the former [13].

- First we are going to use an additive smoothing which consist of considering that every possible element appears δ times more than it actually appears in the corpus, with usually $0 < \delta \leq 1$.

$$P_{\text{add}}(X|Y) = \frac{\delta + c(X,Y)}{\sum_{X'} \delta + c(X',Y)} \tag{5}$$

  where c is the function counting the number of times an element appears in the corpus. This smoothing enable the model to overcome the problem of zero probabilities which often occurs with small training corpora.

- Then, we are going to use a back-off smoothing which consist of using information from a lower order model.

$$P_{\text{back-off}}(X|Y) = \lambda P(X|Y) + (1-\lambda) P(X|Z) \tag{6}$$

  where Z is a subset of Y. For instance, if $P(X|Y)$ is a n-gram, then $P(X|Z)$ could be a $(n-1)$-gram. This smoothing enable the model to overcome the problem of overfitting

2.2 Application to improvisation

In order to test sub-model interpolation for melody generation, we have used a corpus of 50 tunes from the Omnibook [21] composed, played and improvised on by Charlie Parker. We divided this corpus into three sub-corpora:

- a training corpus consisting of 40 tunes and improvisations in order to train the different sub-models,
- a validation corpus consisting of 5 tunes and improvisations in order to optimise the interpolation and smoothing coefficients using cross-entropy minimisation,
- a test corpus consisting of 5 tunes and improvisations.

We decided to use two sub-models :

$$P_1(M_t|X_{1:t}) = P(M_t|M_{t-1})$$

$$P_2(M_t|X_{1:t}) = P(M_t|C_t)$$

where M_t represents the melody at time t, and C_t represents the chord label at time t.

We applied a combination of additive smoothing and back-off smoothing techniques using $P(M_t)$ as a lower order model. Therefore, for the linear interpolation, we have :

$$\begin{aligned} P(M_t|X_{1:t}) = {} & \alpha P(M_t) + \beta U(M_t) \\ & + \lambda_1 P(M_t|M_{t-1}) + \lambda_2 P(M_t|C_t) \end{aligned} \tag{7}$$

where α and β are the smoothing coefficients corresponding respectively to the back-off smoothing and additive smoothing, U is the uniform distribution and λ_1 and λ_2 are the interpolation coefficients. The conditional probabilities are estimated using the counting function c.

	coefficients				cross-entropy
	λ_1	λ_2	α	β	$H(M)$
B+M	**0.582**	**0.129**	**0.289**	**0**	**4.543**
B	0.672	0	0.328	0	4.572
M	0	0.639	0.361	0	4.881
U	0	0	0.998	0.002	5.858

Table 1. Cross-entropy results (bits/note) with linear interpolation. The results are shown for the smooth interpolation of the bigram model and melody/chord model (B+M), then for the bigram model with smoothing (B), then for the melody/chord model with smoothing (M), and finally with the smoothing alone (U) as a point of comparison.

In order to evaluate this model, we used the cross-entropy on the test corpus :

$$H(M) = -\frac{1}{T}\sum_{t=1}^{T} \log_2 P(M_t|X_{1:t}) \,. \tag{8}$$

This metric is in this case equivalent to the KL-divergence up to an additive constant and represents the lack of understanding of the system. Therefore, the lower the cross-entropy, the better the model prediction power.
In Table 1, we present some of the results obtained with linear interpolation. Note that all the results are shown with the same smoothing technique in order to allow a proper comparison. As shown, the model has a better prediction power when using sub-model interpolation. However, the improvement is quite small in term of cross-entropy. This can be explained by the fact that the cross-entropy represents the system's ability to reproduce the test data, while improvisation is not about reproduction but about creativity, and as we said improvisation possibilities are unlimited.

However, informal listening tests show some improvement when using the interpolated model compared to a classic n-gram model. But generated improvisation with just this probabilistic model lack of consistency and of a local organisation. Therefore, we have decided to go further using this type of probabilistic model by combining them with the oracle factor.

3. FACTOR ORACLE EXPLOITING A PROBABILISTIC MODEL

The factor oracle is a structure coming from the field of bioinformatics and language theory [17, 22] that has been widely used in automatic improvisation systems such as OMax [1, 16], ImproteK [5], SoMax [7] or PyOracle [8]. This structure is able to keep the linear aspect of what is being learnt and create links, called suffix links, between places in the memory with a similar context. An example of factor oracle is shown Figure 1.

We designed a system combining the probabilitic model able to take into account the multidimensional aspect of music, with the contextual setting brought by the factor

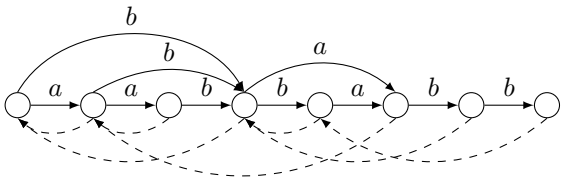


Figure 1. Example of factor oracle constructed on the word $w = aabbabb$. Horizontal solid arrows are the transition, bent solid arrows are the factor links and dashed arrows are the suffix links.

oracle. The idea was to conceive a system creating improvisation in a way closer to a human improviser. We were inspired by this quote from Marilyn Crispell's Elements of Improvisation [23] (written for Cecil Taylor and Anthony Braxton) :

> *The development of a motive should be done in a logical, organic way, not haphazardly (improvisation as spontaneous composition) – not, however, in a preconceived way – rather in a way based on intuition enriched with knowledge (from all the study, playing, listening, exposure to various musical styles, etc., that have occurred through a lifetime – including all life experiences); the result is a personal musical vocabulary.*

First, we create a probabilistic module with all the sub-models we want to take into consideration and the corresponding interpolation and smoothing coefficients necessary to the creation of the global probabilistic model. This module can be trained on a substantial corpus offline, but can also be trained (or updated) online with a musician's playing. In Crispell's quote, this matches with the knowledge acquired through the system's lifetime.
Second, we create an oracle factor for which the construction of states, edges and suffix links only depends on one dimension (usually the melody). The states can represent a single note as in OMax or a musical fragment (for instance a beat) as in ImproteK. In Crispell's quote, this correponds to the logic of the context in which the motive must be developed. The oracle is created online with a musician's playing, or with a corpus (usually smaller than the one used to create the probabilistic module).

The system is now able to improvise music, creating a path in the factor oracle that is guided and enriched by the knowledge from the probabilistic module. At each step, knowing the state the system is in, all the reachable states, and the musical contents in those states, we compute a score for each possible transition corresponding to the interpolation of the sub-models in the probabilistic module. Thus, we are enriching with external knowledge the decision of which edge to follow. We can then normalise the scores to obtain the probabilites of transitions and make a random choice following the resulting probabilities.
Let Att(i) be the set of reachable states from state i follow-

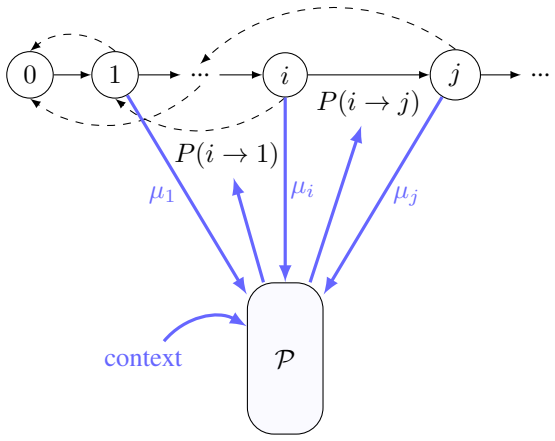


Figure 2. Using a multidimensional probabilistic model $\mathcal{P}$ with an oracle factor. Let us consider that from state i, the only reachable states are state j and state 1. Using the context, μ_1, and μ_i, $\mathcal{P}$ is able to compute a score for the transition from state i to 1. Same thing for the transition from state i to j using the context, μ_i and μ_j. The score are then normalised to get $P(i \to 1)$ and $P(i \to j)$.

ing the heuristics explained in [16] (using suffix links and reverse suffix links for instance). Let $\mu_i = \{\mu_i^M, \mu_i^C, ...\}$ be the musical contents of state i, that is to say the set of musical variables stored in state i during the oracle construction (for instance, μ_i^M represents the musical content's melody of state i). Then, for all $j \in \text{Att}(i)$, the transition probability in the oracle from state i to state j, knowing the past context is :

$$P(i \to j | X_{1:t}) = \frac{P(M_t = \mu_j^M | X_{1:t})}{\sum_{k \in \text{Att}(i)} P(M_t = \mu_k^M | X_{1:t})} \quad (9)$$

In practice, for $X_{1:t}$, we use the musical contents from the previous and current states of the path of the factor oracle. Figure 2 illustrates this process for one step.

4. EXPERIMENTATION

To test the system proposed in the previous part, we generated some improvisations on Charlie Parker's music following three methods.

1. Some improvisations were made with OMax without any probabilistic module. The factor oracle was constructed on one tune (theme and Parker's improvisation).

2. Some improvisations were made with OMax with a probabilistic module. The sub-models considered are an n-gram model over the melody, and a relational model between melody and harmony. The probabilistic module was trained on Charlie Parker's whole Omnibook (50 themes and improvisations), and the factor oracle was constructed on one tune. The Omnibook corpus was created manually using MusicXML and includes both melodic information and chord labels. The idea here is to have a probabilistic module trained on a larger but similar corpus to the tune used for the factor oracle.

3. Some improvisations were made with OMax with a probabilistic module, similarly to the previous one, but the corpus used to train the probabilistic module is a classical music corpus of over 850 non improvised tunes while the factor oracle is constructed on a Charlie Parker tune (theme and improvisation). The classical music corpus was user-generated using MusicXML with both melodic and chord information and was screened for improper chord labels [12]. The idea here is to see how the system performs when trained on a corpus of a different style than the tune used for the factor oracle.

In the second and third method, the probalistic modules were trained using both melodic and harmonic information over all the tunes of each corpus. Three sub-models were used:

$$P_1(M_n | X_{1:n}) = P(M_n | M_{n-1})$$
$$P_2(M_n | X_{1:n}) = P(M_n | C_n)$$
$$P_3(C_n | X_{1:n}) = P(C_n | C_{n-1})$$

where n is an index over the note of the melody. M_n is the n^{th} notes of the melody, and C_n is the chord played over M_n.
Due to the nature of our dataset, we chose to use a small amount of sub-models and very simple one as a proof of concept. Better results would be expected with more sub-models (as mentioned in section 2.1) but would require more complete data.

For each method, 15 improvisations were generated using 3 Charlie Parker tunes as reference : Au Private, Donna Lee and Yardbird Suite.
The generated improvisations can be listened online at `members.loria.fr/evincent/files/smc16` and the MusicXML Omnibook corpus can be found at `members.loria.fr/evincent/files/omnibook`.

First of all, the most significant difference seems to be the harmonic stability appearing while using a probabilistic module trained with either the Omnibook or a classical music corpus. The improvisations generated using these methods seem to follow a harmonic framework, while the factor oracle is only constructed with the melody. For instance, this can be heard on the first example of Au Private. Second, when the probabilistic module is trained on a classical music corpus, while the harmonic stability is stronger, Charlier Parker's musical language looses its distinctiveness, as if the harmonic aspect was too strong a constraint. For instance, this can be noticed on the third example of Yardbird Suite. This comforts our initial idea that using a multidimensional training over an appropriate corpus enables our system to generate improvisations closer to a specific style.
Furthermore, according to listeners, the improvisations with a probabilistic module are more diverse, fluid and creative

than the simple oracle one. This is in part because the combination of dimensions and the smoothing provide escape mechanisms from usual mono-dimensional attractors (the obsessive jingle phenomenon due to high conditional probabilities and overfitting). For instance, this can be clearly heard in the first example of Donna Lee.

These results are encouraging. We only tested this system using melodic and harmonic relations, yet we can already hear a significant improvement on how the improvisations are guided through the factor oracle. This system could be extended to represent other interdimensional relations, in particular rhythm, beat phase and dynamic, with more detailed data from live playings, and therefore can be used for any style of music.
Moreover, this system's modularity makes it very adaptable, and could be integrated in other existing systems :

- A probabilistic module could be integrated in ImproteK [5], where the evolution of one dimension is predefined in a scenario. This would add some smoothing in ImproteK's improvisation and therefore expand its expressiveness.

- Similarly, a probabilistic module could be integrated in SoMax [7] where some of the context would come from active listening.

- Finally, this system could be adapted for PyOracle [8] using an interpolation where the dimensions are actually audio features.

5. CONCLUSIONS

We have shown the musical potentialities of the combination of probabilistic models with the factor oracle. This creates a system able to follow the contextual logic of an improvisation while enriching its musical discourse from multidimensional knowledge in a closer way to a human improviser. On the one hand, the probabilistic models enable the system to be trained on a multidimensional sequence and to take the relations between dimensions into account. They also profit from advanced smoothing and optimisation techniques which make them an efficient way to represent the musical knowledge acquired through a lifetime by a musician. On the other hand, the factor oracle is an efficient data structure able to represent the logic of a musical context. This system shows good potential to perform a better navigation in the factor oracle, generating improvisations closer to the desired style. Moreover, this system could be easily adaptated to other existing systems (ImproteK, SoMax, PyOracle...), potentially improving their results.

Acknowledgments

This work is made with the support of the French National Research Agency, in the framework of the project DYCI2 "Creative Dynamics of Improvised Interaction" (ANR-14-CE24-0002-01), and with the support of Region Lorraine.

6. REFERENCES

[1] G. Assayag and S. Dubnov, "Using factor oracles for machine improvisation," *Soft Computing*, vol. 8-9, pp. 604–610, 2004.

[2] D. Conklin, "Music generation from statistical models," in *Proceedings of the AISB Symp. on Artificial Intelligence and Creativity in the Arts and Sciences*, 2003, pp. 30–35.

[3] F. Pachet and P. Roy, "Markov constraints: steerable generation of Markov sequences," *Constraints*, vol. 16, no. 2, pp. 148–172, March 2011.

[4] S. Dubnov, G. Assayag, O. Lartillot, and G. Bejerano, "Using machine-learning methods for musical style modeling," *IEEE Computer*, vol. 10, no. 38, pp. 73–80, 2003.

[5] J. Nika and M. Chemillier, "ImproteK, integrating harmonic controls into improvisation in the filiation of OMax," in *Proceedings of the International Computer Music Conference*, 2012, pp. 180–187.

[6] J. Nika, J. Echeveste, M. Chemillier, and J.-L. Giavitto, "Planning human-computer improvisation," in *Proceedings of the International Computer Music Conference*, 2014, pp. 330–338.

[7] L. Bonasse-Gahot, "An update on the SoMax project," IRCAM, Tech. Rep., 2014.

[8] G. Surges and S. Dubnov, "Feature selection and composition using pyoracle," in *Proceedings of the 2nd International Workshop on Musical Metacreation*, 2013.

[9] A. Donze, S. Libkind, S. A. Seshia, and D. Wessel, "Control improvisation with application to music," EECS Department, University of California, Berkeley, Tech. Rep. UCB/EECS-2013-183, November 2013.

[10] D. Conklin and I. H. Witten, "Multiple viewpoint systems for music prediction," *Journal of New Music Research*, vol. 1, no. 24, pp. 51–73, 1995.

[11] R. P. Whorley, G. A. Wiggins, C. Rhodes, and M. T. Pearce, "Multiple viewpoint systems: Time complexity and the construction of domains for complex musical viewpoints in the harmonisation problem," *Journal of New Music Research*, no. 42, pp. 237–266, 2013.

[12] S. A. Raczyński, S. Fukayama, and E. Vincent, "Melody harmonisation with interpolated probabilistic models," *Journal of New Music Research*, vol. 42, no. 3, pp. 223–235, 2013.

[13] S. F. Chen and J. Goodman, "An empirical study of smoothing techniques for language modeling," Harvard University, Tech. Rep. TR-10-98, 1998.

[14] M. I. Bellgard and C. P. Tsang, "Musical networks," N. Griffith and P. M. Todd, Eds. MIT Press, 1999, ch. Harmonizing Music the Boltzmann Way, pp. 261–277.

[15] G. Bickerman, S. Bosley, P. Swire, and R. M. Keller, "Learning to create jazz melodies using deep belief nets," in *Proceedings of the International Conference on Computational Creativity*, 2010, pp. 228–236.

[16] G. Assayag and G. Bloch, "Navigating the oracle : A heuristic approach," in *Proceedings of the International Computer Music Conference*, 2007, pp. 405–412.

[17] C. Allauzen, M. Crochemore, and M. Raffinot, "Factor oracle : A new structure for pattern matching," *SOFSEM'99, Theory and Practice of Informatics*, pp. 291–306, 1999.

[18] F. Jelinek and R. L. Mercer, "Interpolated estimation of Markov source parameters from sparse data," in *Pattern Recognition in Practice*, 1980, pp. 381–397.

[19] D. Klakow, "Log-linear interpolation of language models," in *Proceedings of the 5th International Conference on Spoken Language Processing*, 1998, pp. 1695–1698.

[20] S. F. Chen, L. Mangu, B. Ramabhadran, R. Sarikaya, and A. Sethy, "Scaling shrinkage-based language models," in *Proceedings of the Automatic Speech Recognition & Understanding*, 2009, pp. 299–304.

[21] C. Parker and J. Aebersold, *Charlie Parker Omnibook*. Alfred Music Publishing, 1978.

[22] A. Lefebvre and T. Lecroq, "Computing repeated factors with a factor oracle," in *Proceedings of the 11th Australasian Workshop On Combinatorial Algorithms*, 2000, pp. 145–158.

[23] M. Crispell, "Elements of improvisation," in *Arcana : Musicians on Music*, J. Zorn, Ed., 2000, pp. 190–192.

EXPLORING MOMENT-FORM IN GENERATIVE MUSIC

Arne Eigenfeldt
School for the Contemporary Arts, Simon Fraser University, Canada
arne_e@sfu.ca

ABSTRACT

Generative art is art created through the use of a system. A unique and distinguishing characteristic of generative artworks is that they change with each run of the system; in the case of generative music, a musical composition that re-explores itself, continually producing alternative versions. An open problem in generative music is large-scale structure: how can generative systems avoid creating music that meanders aimlessly, yet doesn't require strict architectural forms into which it is forced inside? *Moments* is a generative installation that explores Moment-form, a term Stockhausen coined to describe (his) music that avoids directed narrative curves. Through the use of musebots – independent musical agents – that utilise a *parameterBot* to generate an overall template of "moments", the agents communicate their intentions and coordinate conditions for collaborative machine composition.

1. INTRODUCTION

Generative art refers to art that has been created with the use of a system. Such artist-designed systems make decisions that are normally made by the artist. Galanter points out that this type of art has a long tradition, and such approaches may be "as old as art itself" [1]. Metacreation is the contemporary approach to generative art, and looks at all aspects of the creative process and their potential for systematic exploration through software. This field is populated by a diverse group of people – psychologists, art theorists, cognitive scientists, artificial intelligence researchers, machine learning specialists, and, perhaps most importantly, artists. Musical Metacreation (MuMe) has proven to be a fertile creative domain for composers exploring new avenues of production.

1.1 Musical Metacreation (MuMe)

Galanter's definition of generative art [1] – "any art practice where the artist uses a system… which is set into motion with some degree of autonomy *contributing to or resulting in a completed work of art*" (italics mine) – assumes a fully finished artwork. Furthermore, implicit in this definition is that the system may involve human interaction, in that the system need only *contribute* to the final work. As such, human involvement, whether through algorithm design, direct control by an operator, or interaction with a live human performer, has remained an active presence in the dynamic generation of music.

2. MUSICAL FORM

One reason for continued human interaction in MuMe is that generating entire musical compositions entails the development of musical form, a highly complex task [2]. Form involves the complex interaction of multiple musical structures in order to logically organise the work's progression in time. Strategies are required to organise these structures so as to "provide reference points for the listener to hold on to the piece, otherwise it may lose its sense of unity" [3].

Musical form is "the unique result of the deployment of particular materials and processes" [4]. Form is the consequence of the relationships between the various structures within the music – themselves methods of organising pitch, onsets, volume, and timbre; the surface relationships within the music – i.e. the selection and resulting relationships between individual musical objects at a given point in time – can be considered its design [after 5].

Kramer suggests that one difficulty in conceptualising form is due to its inherent role in organising time. While theories exist dealing with rhythm and meter, "more difficult to discuss are motion, continuity, progression, pacing, proportion, duration, and tempo" [6], all aspects to do with musical form. Schoenberg expressed the complexity of form, stating that it requires "logic and coherence" in order for a musical composition to be perceived as being comprehensible, but that its elements also should function "like those of a living organism" [7]. The difficulty for composers is achieving a balance between strict structures that appears logical, with an organic element that engenders surprise.

2.1 Generating Form

This difficulty is multiplied exponentially when applied to generative music: how can one codify structural decisions when many of these decisions are aesthetic in nature? For example, interactive systems allow the composer to determine when to move to the next section, or when to alter a process, based upon choices informed by context – how long has the current section been going on? – and aesthetics – is the material starting to lose interest? Different surface features (i.e. the musical context) will engender different decisions; codifying such pro-

cesses suggests the need for computational aesthetic evaluation, a highly complex notion that remains an open problem [8].

Nevertheless, MuMe has produced a variety of heuristic solutions to the problem of generating structure. Previous computational models of musical structure include Cypher [9], Experiments in Musical Intelligence [10], GESMI [11], the use of statistical prediction [e.g. 12], the use of machine learning techniques [e.g. 13], and agent negotiation [e.g. 14].

Two contrasting methodologies have dominated the field: top-down versus bottom-up methods. The former relies upon architectural models that may be pre-generated in varying degrees: Cope's use of SPEAC is one such example [15], as well as GESMI's structures derived from a corpus of dance music [11]. Bottom-up approaches, which Boulez described as "form from material" [16], entail methods of self-organisation [17]. Narmour [18] provides a useful discussion on the interaction and opposition of these two approaches in musical composition.

3. NON-TELEOLOGICAL MUSICAL FORM

A clear difficulty in generating structure, as pointed out above, results from adopting the complex procedures (i.e. goal-directedness) from functional tonality. Tonality, the overriding organisational structure of music from 1600-1900 (and continuing today in many styles), is the music of continuity and motion. With the dissolution of functional tonality in the early 20th century, composers were forced to search for other methods of goal-directed motion. While composers such as Schoenberg attempted to construct new methods for continuity – provoking the adage "new wine in old bottles" – Stravinsky investigated non-developmental methods as early as *Le Sacre* (1913), and discontinuity in *Symphonies of Wind Instruments* (1920)

Similarly, Debussy (*Jeux*, 1913), Webern (*Symphony*, second movement, 1928), Varèse (*Ionisation*, 1931), and Messiaen (*Quatuor pour la fin du temps*, 1941) composed music that seemingly avoided forward motion and favoured discontinuity between sections. By the 1960s, these ideas were overtly adopted by Cage in his time frames, Reich in his gradual processes, Glass in his additive processes, and, most importantly, Stockhausen in his Moment-form [19]. As Smalley suggests, "Moment-form is the only really new, linguistically independent and therefore generally applicable formal concept to have arisen since 1945" [20].

3.1 Stockhausen and Moment-form

First fully explored musically in *Momente* (1962-69), Stockhausen described the potential of Moment-form in *Texte zur Musik* in 1963:

> *"A given moment is not merely regarded as the consequence of the previous one and the prelude to the coming one, but as something individual, independent and centered in itself, capable of existing on its own."* [19].

Momente continued to explore musical structure from a non-pitch dominated perspective that began with *Kontakte* (1958–60). Unlike the total serialist explorations of the previous decade, which still relied upon pitch for structural elements, Stockhausen's new model derived structure "from the totality of possibilities inherent in the diverse materials which the composer brings together for each particular work" [20].

3.1.1 Requirements for Moment-form

A moment is comprised of a static entity – for example, a single harmony; moments avoid development and goal-directed behaviour, although the potential for processes to provide variation in the surface design is possible. Subsequent moments are contrasting, often dramatically, with one another, as their internal organisation and concerns must be different; as a result, changes between moments result in what Kramer refers to as discontinuity [21]. Using Salzer's definitions, described earlier, each moment contains its own structure; it can consist of a great deal of surface variation, as long as the variation does not contribute to goal-directed behaviour; the resulting combination of contrasting moments results in the final Moment-form.

Momente also uses a mobile feature in which the individual moments can be reordered in different ways, which Stockhausen refers to as polyvalent: "A composer is no longer in the position of beginning from a fixed point in time and moving forwards from it; rather he is moving in all directions within a materially circumscribed world" [20]. Because different moments must be self-contained, and are defined by contrasting states, the order of moments should not matter; however, polyvalence – the re-ordering of moments between performances – is not a requirement. As Kramer suggests, "the order of moments must *appear* arbitrary for the work to conform to the spirit of moment form" [21].

Stockhausen separates beginning and ending, from starting and stopping: the former pair he equates with dramatic (closed) forms, the latter pair with open moment forms. Although compositions need to begin and end for practical reasons, "a proper moment form will give the impression of starting in the midst of previously unheard music, and it will break off without reaching any structural cadence, as if the music goes on, inaudibly [21]. This concepts gives rise to a feeling of "endlessness" within Moment-form.

3.1.2 Proportion

Kramer states that in Stravinsky's precursor to Moment-form, specifically in *Symphonies of Wind Instruments*, the motivic and tempo consistency within a section results in its self-containment and staticism; progression in the music takes place between, rather than within, moments [21]. The relative length of each moment is an integral aspect of a work's formal success, as proportional length is one of the only remaining principles of formal coherence within Moment-form [21]. Kramer analyses a number of works, demonstrating the reliance on, for example,

3:2 time ratios (which also can be considered the golden ratio, as well as the Fibonacci series)[22] between moments in the music of Stravinsky [21]; Maconie does the same for Stockhausen [23], while Parks describes the music of Debussy in terms of its proportional relationships and discontinuity [24].

3.2 Moment-form in ambient music

Kramer was considering art music and the evolving tradition leading up to Stockhausen, at almost the same time that an ambient aesthetic emerged outside of concert music. While ambient music can itself be traced back to Satie's furniture music [25], Eno popularised the music with seminal recordings in the late 1970s which were meant "to accommodate many levels of listening attention without enforcing one in particular" [26]. The result was music that avoided dramatic change and motion, and tended to preference continuity and stasis. Unlike Kramer's notion of Moment-form that relies upon discontinuity between moments for an overall structural form, ambient music most often contains only a sin[i]gle moment.

Adopted by the electronic dance music community in the 1990s as a "chill-out" alternative to its beat-oriented music, artists such as Aphex Twin (*Selected Ambient Works Volume II*, 1994), Boards of Canada, and KLF (*Chill Out*, 1990) produced music that often lacked strong beats (or at least the ever-present drumbeat of contemporaneous dance music of the time), and, more importantly in our case, repetitive formal structures [27].

Ambient electronica, and its contemporary offspring, dark- and post-ambient, continues to avoid discontinuity, preferring a relative consistency in harmony and timbre: although gestures may enter and exist – sometimes resulting in separate sub-moments in doing so – the amount of significant timbral change is relatively limited.

Such consistency underlines an integral aspect of Moment-form: staticism. Kramer poses the question "is musical staticism an experiential possibility? How long must it go on before the listener gives up expectation of change and enters a static mode of perception? The answer seems to depend on the *richness of the unchanging sound.*" [21]. In the case of contemporary ambient electronic music, a richness of timbre is perhaps its most obvious feature.

3.2.1 Christopher Bissonnette

One contemporary recording artist whose work, perhaps inadvertently, can be considered as utilising Moment-form is Christopher Bissonnette.

Bissonnette's 2005 album *Periphery*[ii] contains 7 tracks, ranging in length from just under 5 minutes to almost ten minutes, each of which can be considered as a single moment, or several related sub-moments. The music is slow moving, lacking any obvious pulse, and uses drones without harmonic change to create a feeling of staticism.

Figure 1 presents a sonogram of the first track from the album, *In Accordance*, whose duration is almost nine minutes. The analysis displays a clear delineation in frequency space in the 4 moments (or perhaps, sub-moments). The first moment (A) is comprised of a sustained 841 Hz pitch accented with a percussive "ping", which has a repetition rate of 7.2 Hz (8 beats at approximately 57 bpm). Typical of the album, Bissonnette's *oeuvre*, and the larger genre of post-ambient music, the track fades in; however, rather than slowly introducing constituent elements, the gestures seem to already exist, and are discretely brought to our attention. In Stockhausen's terminology, the work has started, but its beginning was long before.

After approximately one minute twenty seconds (B), the listener becomes aware of a new gesture in a higher frequency range: related (it is also percussive and a similar piano-like timbre), but comprised of a slightly different repetition cycle (1.8 Hz, or 32 bpm). Again, typical of Bissonnette, the two frequencies are related – a Just major sixth; the entire work explores the relationship between these two intervals without extending to a larger pitch set.

The introduction of lower frequencies at (C) is slightly more abrupt, seemingly causing one of the original frequency bands to become much less perceptible (although not disappear). A pair of high frequency beating gestures – centered around 2631 and 5386 Hz – as well as transient clicks (suggesting analog record noise) are introduced. The extreme high and low frequencies fade (A + B), revealing the original frequencies of the opening, without the percussive elements.

The proportions of the four sections, despite lacking clear divisions, have durational relationships that, while not exact, demonstrates Kramer's axiom of proportion being a seminal formal element in Moment-form (see Table 1). Although the final section may be considered as a return, it is perceived more as a reminder of earlier textures that never actually disappeared. Note that Stockhausen prohibits recapitulation in his articles, but return does appear in his music.

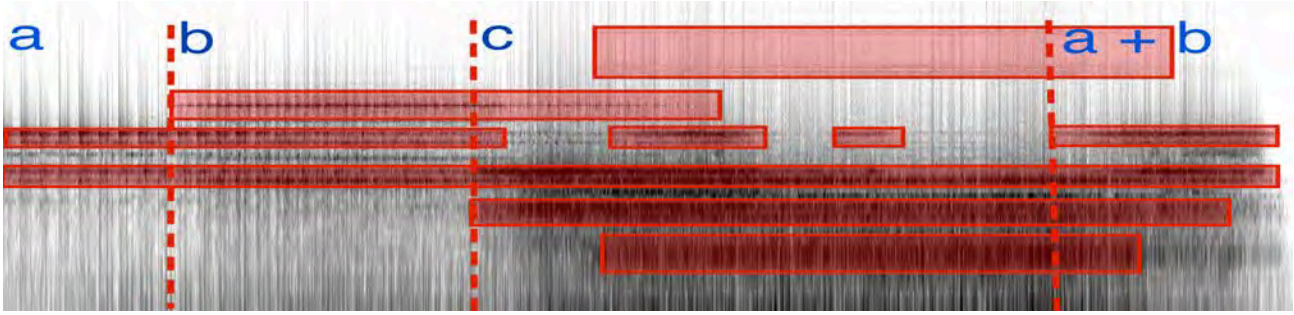


Fig. 1. Sonogram of Christopher Bissonnette's *In Accordance*. Frequency bands are highlighted in red; sub-moments between dotted lines, are lettered.

Section	Start time	Duration (sec)	Ratio
A	0:00	80	1.0
B	1:20	150	1.875
C	3:50	320	4.0
A + B	7:50	120	1.5

Table 1. Proportions in Bissonnette's *In Accordance*

4. GENERATING MOMENT-FORM

As discussed in section 2, generating large-scale structure is a complex task, and remains an open-problem in musical metacreation [14]. Several aspects of Moment-form are thus compelling models for formal generation.

4.1 Generative Potential

4.1.1 Moments as parametric containers

The notion that individual moments should be comprised of unique organisational methods suggests generative potential. Smalley proposes that a composer utilising Moment-form "must be aware of all the potentialities of his material before he actually begins to notate the score", which is often how designers of generative structures approach their material. He suggests that such organisational requirements account for Stockhausen's "obsessive interest in the categorising and pre-compositional ordering of his basic material" [20]. Similarly, generative structures often involve parameterisation of a great deal of musical features, and individual moments can be delineated by unique constraints upon these methods.

4.1.2 Staticism

The consistency of features, to the point of staticism, is an interesting alternative to the model of continual evolution. While most contemporary music that can be considered "tonal" does not rely upon harmony to provide large-scale structure in the same way that composers of the 19th century used it in their goal-directed motion, it does, nonetheless, often exploit cyclical harmonic patterns to outline meso-structures and phrases. In these cases, higher-level forms are created by varying the harmonic cycles, generating typical song-forms (i.e. verse-chorus) or dance forms (A B C D permutations) [28]. Avoiding harmonic movement at the surface level avoids the need for harmonic change at the structural level, a device employed by contemporary ambient artists in their avoidance of all harmonic change. Instead, surface variation – and listener interest – is maintained by generating processes that explore timbral alteration (e.g. filter sweeps) or highlighting different pitch classes in a unvarying pitch set.

Generative practices are already being used in ambient electronica, although not always fully acknowledged. Marsen Jules (Martin Juhls), an exception to this, is a self-proclaimed generative electronic musician who has released ten albums whose music "modulates on the basis of strict rules…varying over and over and thus emerges from the very moment itself" [29], using techniques inspired by Steve Reich and Brian Eno. Juhls demonstrates a further trait of contemporary ambient artists' take on moment-form, in that he is interested in consistency, and avoids any contrast that may suggest discontinuity (personal communication).

4.2 Musebots

There have been many successful MuMe production systems that have generated complete musical works, and therefore generated long-term musical structure. As MuMe does not exclude human-machine interaction, these systems have tended to rely upon human-machine partnerships. More recently, creative research has been undertaken to collaboratively explore autonomous machine-machine generative systems through the use of specifically designed collaborative musical agents. Musebots [30] are autonomous musical agents that interact in performance, messaging their current states in order to allow other musebots to adapt. Recent musebots have been developed that broadcast their intentions, and not just their current state, thereby allowing other musebots to modify their own plans.

A particularly exciting aspect of musebots involves the notion that developers must decide *how* the musebots should interact, and *what* information is necessary to produce meaningful musical interaction. Musebots offer the potential to create complex musical surfaces and structures in which the organisation cannot be pointed to a single clever programmer. Naturally, concepts of formal design have been raised: initial musebot ensembles followed either a self-organising model, or a reactive model in which one musebot "took the lead" in determining sectional change. Musebot ensembles have thus far avoided the requirement of large-scale formal structures by limiting their performances to five to seven minute compositions.

5. MOMENTS

Musebots allow for a collective, collaborative approach to generative music production; however, as they require a consensus in what information is to be communicated, a singular design approach has proven to be more successful in exploring a particular compositåional perspective. *Moments* is an ongoing installation using musebots created only by the author, in which continuous Moment-forms are generated, and subsequently explored sonically.

With each new composition, a *parameterBot* is initially launched that determines the next composition's ensemble, selecting from pre-curated combinations of musebots. Each musebot has preferred generative tendencies, including timbral properties (i.e. synthesis techniques), frequency ranges (i.e. low versus high frequencies), and potential for background (i.e drones) versus foreground (i.e. more transient) gestures.

Once a new ensemble of musebots has been launched, the *parameterBot* decides upon the overall compositional duration, and the number of sections (moments) contained within the work. Durations for each section are generated proportionally, with subsequent section being either in 2:3 or 3:2 durational relationships to the current section.

Parameter values for a variety of features are also calculated. These values can be considered as mean values within a constrained random range. Since the range is also generated and communicated, the combination of the two values is essentially a tendency mask for the feature over the duration of the section [31]. At the moment, features that are generated include:

- speed (tempo);
- activityLevel (relative density of events);
- voiceDensity (number of active voices);
- complexity (regarding harmony, melodic shape, syncopation);
- volume;
- consistency (amount of surface level change)
- pitch (mean pitch).

Messages are sent for each parameter within each section, with a start and end value for the section; thus, the same value for start and end assumes a consistent level during that section, while two different values assumes a linear movement between the two values (see Figure 2).

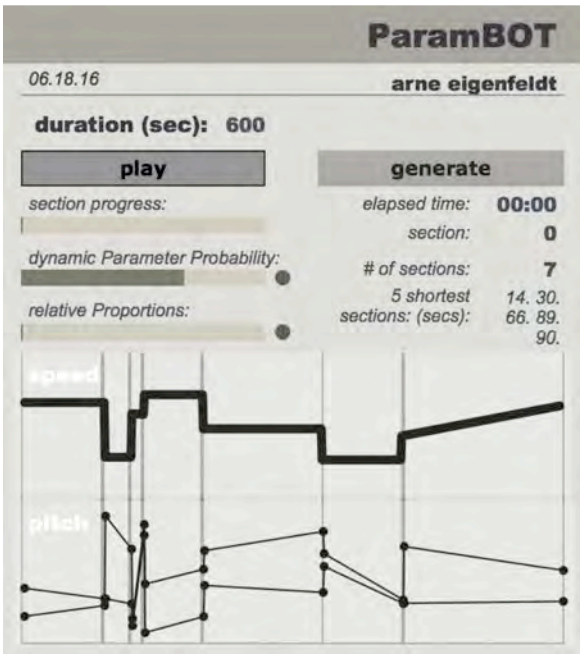


Fig. 2. *parameterBot* displaying a ten minute composition containing seven moments/sections, with the speed (thick line) and pitch parameters with surrounding tendency masks (thin lines) shown.

Once the values have been messaged to the musebot agents, the musebot Conductor is initiated, which provides a continuously running clock, beginning at time zero. Musebots use the clock time to determine their location in the overall form as well as the current section.

Surface features, including when to play, and how often events occur, are negotiated by the musebots themselves. Given the prescribed parameters – for example, activityLevel and voiceDensity – musebots continually communicate their current state, and to some extent, their future actions, thereby allowing other musebots to adjust their own activities.

The potential for autonomous musical agents to negotiate a requested feature value was explored within *Kinetic Engine* [32], albeit limited to rhythmic density. In that case, agents contained enough musical intelligence to generate correct musical responses that would vary depending upon other agent actions, as well as new requests from a Conductor (a human performer). The *Coming Together* series [33] explored autonomous requests without human interaction; however, overall form was dependent upon self-organisation, and not a request in itself. One goal for *Moments* is to use a Listener musebot that collects individual musebot activity, and compares the current states to the request from the *parameterBot;* the Listener bot would then send its own messages, informing the ensemble whether to adjust current parameters (i.e. more cowbell) depending upon inadequately negotiated targets.

6. FUTRE WORK AND CONCLUSION

Moments is in an early stage of what is clearly going to be a long-term project. Like all generative music, finding the "sweet spots" for parameter ranges, determining which parameters to automate, and the complex interaction between parameters, will require a great deal of listening to the output of the system. As Marsen Jules states, "Usually when I create this kind of music I listen to the set up for hours to explore all details that happen in the setting" [29].

Generative music, and its contemporary offspring, Musical Metacreation, offers composers the opportunity to explore processes and systems that create continually changing music. While generating large-scale structure remains an open problem in MuMe, exploring Moment-form avoids the (potentially unnecessary) complexities of traditional formal models, utilising instead the only new formal model invented since the dissolution of tonality over one hundred years ago: Moment-form. By actively seeking staticism, eschewing beginnings and endings, avoiding harmony as a structural feature, and emphasising timbre and frequency space as primary structural avenues, Moment-form offers a pragmatic alternative to traditional goal-based music, especially generative music.

Acknowledgments

The author wishes to acknowledge the Social Science and Humanities Research Council of Canada for continued funding of this research; Jim and Justine Bizzocchi of the Generative Media Project for an opportunity to explore musebots and formal generation in an artistic context; and Ollie Bown, Andrew Brown, and Toby Gifford for the extended work on musebots and ideas regarding generating form within that framework.

7. REFERENCES

[1] P. Galanter, "What is Generative Art? Complexity theory as a context for art theory," in Generative Art Conference (GA 2003), Milan, 2003.

[2] W. Berry, Form in Music (Vol. 1). Prentice-Hall, 1966.
[3] E. Miranda, Composing Music with Computers. CRC Press, 2001.
[4] A. Whittall, "Form," Grove Music Online. Oxford Music Online. Oxford University Press, accessed Feb. 1, 2016.
[5] F. Salzer, Structural hearing: Tonal Coherence in Music. Courier Corporation, 1962.
[6] J. Kramer, The Time of Music New Meanings, New Temporalities, New Listening Strategies. Schirmer Books, 1988.
[7] A. Schoenberg and L. Stein, Fundamentals of Musical Composition. Faber & Faber, 1970.
[8] P. Galanter, "Computational aesthetic evaluation: past and future," in Computers and Creativity, Springer Berlin, 2012, pp. 255–293.
[9] R. Rowe, Interactive Music Systems: Machine Listening and Composing. MIT Press, 1992.
[10] D. Cope, Experiments in Musical Intelligence. A-R Editions, 1996.
[11] A. Eigenfeldt, "Generating Electronica - A Virtual Producer and Virtual DJ," in Proceedings of the 9th ACM Conference on Creativity & Cognition, Sydney, 2013, p. 396.
[12] D. Conklin, "Music Generation from Statistical Models", in AISB 2003 Symposium on Artificial Intelligence and Creativity in the Arts and Sciences, Aberystwyth, 2003, pp. 30–35.
[13] B. Smith and G. Garnett, "Improvising Musical Structure with Hierarchical Neural Nets," in Proceedings of the Artificial Intelligence and Interactive Digital Entertainment Conference, Palo Alto, 2012, pp. 63–67.
[14] A. Eigenfeldt, "Generating Structure – Towards Large-Scale Formal Generation," in Proceedings of the Artificial Intelligence and Interactive Digital Entertainment Conference, Raleigh, 2014, pp. 2–9.
[15] D. Cope, The Algorithmic Composer. A-R Editions, 2000.
[16] P. Boulez, D. Noakes, and P. Jacobs, "Alea," in Perspectives of New Music 3(1), 1964, pp. 42–53.
[17] T. Blackwell and M. Young, "Self-organised music", in Organised Sound 9(02), 2004, pp. 123–136.
[18] E. Narmour, "The Top-down and Bottom-up Systems of Musical Implication: Building on Meyer's Theory of Emotional Syntax," in Music Perception: An Interdisciplinary Journal 9(1), 1991, pp. 1–26.
[19] K. Stockhausen, "Momentform: Neue Beziehungen zwischen Aufführungsdauer, Werkdauer und Moment," in Texte zur Musik 1, 1963, pp. 189–210.
[20] R. Smalley, "'Momente': Material for the Listener and Composer: 1," in The Musical Times 115 (1571), 1974, pp. 23–28.
[21] J. Kramer, "Moment form in twentieth century music," in The Musical Quarterly 64(2), 1978, pp. 177–194.
[22] J. Kramer, "The Fibonacci series in twentieth-century music," in Journal of Music Theory 17(1), 1973, 110–148.
[23] R. Maconie, The Works of Karlheinz Stockhausen. Oxford University Press, 1976.
[24] R. Parks, The Music of Claude Debussy. Yale University Press, 1989.
[25] R. Orledge, "Understanding Satie's Vexations", in Music & Letters 79(3), 1998, 386–395.
[26] B. Eno, *Ambient Music.* http://www.iub.edu/~audioweb/T369/eno-ambient.pdf, accessed April 11, 2016.
[27] P. Shapiro, Modulations: a history of electronic music: throbbing words on sound. Distributed Art Publishers, 2000.
[28] A. Eigenfeldt, and P. Pasquier, "Evolving structures for electronic dance music," in Proceedings of the 15th annual conference on Genetic and evolutionary computation (GECCO 2013), 2013, pp. 319–326.
[29] M. Juhles, *Marsen Jules*. http://www.marsenjules.de/, accessed April 18, 2016.
[30] O. Bown, B. Carey, and A. Eigenfeldt, "Manifesto for a Musebot Ensemble: A platform for live interactive performance between multiple autonomous musical agents," in Proceedings of the International Symposium of Electronic Art (ISEA 2015), 2015.
[31] B. Truax, "Real-time granular synthesis with a digital signal processor," in *Computer Music Journal* 12(2), 1988, pp. 14-26.
[32] A. Eigenfeldt, "Emergent rhythms through multi-agency in Max/MSP," in Computer Music Modeling and Retrieval. Sense of Sounds. Springer Berlin Heidelberg, 2007, pp. 368–379.
[33] A. Eigenfeldt, "Coming together: negotiated content by multi-agents," in Proceedings of the ACM International Conference on Multimedia, Florence, 2010, pp. 1583–1586.

[ii]https://open.spotify.com/album/0YzwNo517SNbNMITxRIt6p

TSAM: A TOOL FOR ANALYZING, MODELING, AND MAPPING THE TIMBRE OF SOUND SYNTHESIZERS

Stefano Fasciani
Faculty of Engineering and Information Sciences
University of Wollongong in Dubai
stefanofasciani@stefanofasciani.com

ABSTRACT

Synthesis algorithms often have a large number of adjustable parameters that determine the generated sound and its resultant psychoacoustic features. The relationship between parameters and timbre is important for end users, but it is generally unknown, complex, and difficult to analytically derive. In this paper we introduce a strategy for the analysis of the sonic response of synthesizers subject to the variation of an arbitrary set of parameters. We use an extensive set of sound descriptors which are ranked using a novel metric based on statistical analysis. This enables the study of how changes to a synthesis parameter affect timbral descriptors, and provides a multi-dimensional model for the mapping of the synthesis control through specific timbre spaces. The analysis, modeling and mapping are integrated in the Timbre Space Analyzer & Mapper (TSAM) tool, which enables further investigation into synthesis sonic response and on perceptually related sonic interactions.

1. INTRODUCTION

The timbre generated by a sound synthesis algorithm depends on the values assigned to the variable parameters, typically user configurable. Regardless of the synthesis method, the relationship between control and perceptual features of the resultant sound is generally weak [1] and difficult to determine. Modern synthesis algorithms present a wide timbre range and a high dimensional control space. The timbre, which is central in modern sonic arts, has high dimensionality as well [2] and a blurry scientific definition [3]. For designers of sonic interactive systems and of musical instruments, knowing the parameter-to-timbre relationship supports the implementation of the intended sonic response. For sound designers and performers this knowledge eases the development of control intimacy [4]. Also, this insight can help in improving the expressivity of musical instruments by reducing the control dimensionality while broadening the timbral response. The heuristic estimation of the parameter-to-timbre causality is workable, but is subjective and inaccurate. This task is challenging due to nonlinearities and correlations in the synthesis process, especially when a large set of variable parameters are involved.

We address this issue by proposing a systematic and generic method to analyze the timbre in relation to the synthesis variables. The collected data is then processed by computing a quality metric for each sound descriptor, composed of four weighted components, each representing a specific statistical characteristic. Additionally, quality metrics for synthesis parameters are provided as well. This information can be used in designing the mapping of musical gestures to the synthesis control, providing a tighter causal link with the timbral response of the system. The tool we present here, the Timbre Space Analyzer & Mapper (TSAM), integrates these functionalities and supports implementation of few-to-many lossless mappings [5], through an intermediate timbre-related layer [6]. The tool, after analyzing the sonic response of the synthesizer, computes a reduced timbre-to-parameter model, which supports real-time interaction with the sound synthesizer. In particular, we integrate an extension of the modeling and mapping strategy we introduced in [7], highlighting the enhancement achieved when considering the quality metric for selecting the descriptor for mapping purposes.

The TSAM is a flexible tool, exposing internal computation settings and options on a Graphical User Interface (GUI), which supports a range of applications and aims. The perceptual characteristics of synthesis method can be studied, characterized, and compared numerically or graphically. The relationship between timbre, spectrum and different musical scales can be investigated [8]. Different mapping approaches for musical instrument can be explored and compared. The rest of this paper is organized as follows. In Section 2 we describe the synthesis analysis procedure and present the quality metric for descriptors and parameters. Section 3 provides a summary of the timbre space mapping strategy. The TSAM implementation is detailed in Section 4. Finally, Section 5 concludes with discussion and future works.

2. TIMBRE RESPONSE ANALYSIS

Understanding the sonic variation resulting by tweaking parameters is common when getting familiar with a sound synthesizer. Different users may have distinct in-

tents. Sound designers aim at synthesizer configurations generating the their desired sound, whereas performers and instrument builders look at a mapping that yields sonic expressivity. Synthesizers generally feature a large number of controllable parameters, representing the synthesis algorithm variables. In analog synthesizers, each parameter can theoretically assume an infinite number of values, while in digital (or software) synthesizer we have more than 4 billion possible values if considering single-precision implementations (32 bit). Synthesizers interfaced using the MIDI protocol allow only up to 128 distinct values per parameter (7 bit), despite the resolution of the internal circuitry. However with only three MIDI controlled parameters we have more than 2 million (2^{21}) different parameter permutations or unique synthesis states. This combinatorial explosion limits the feasibility of a comprehensive analysis of the all timbre resultant from each of these states.

Limiting the dimensionality of the parameter space allows coping with the large number of synthesis states to analyze, laving only a few variable parameters and fixing the remaining to specific values. In this case the timbre analysis is limited to a subset of the entire parameter space, which is a scenario equivalent to users tweaking only a few parameters of a synthesis configuration (or preset). To further reduce the number of states to analyze we use the principle of spatial locality: states close in the parameter space generate similar timbres. Therefore we can sample the parameter space with a larger step size, and eventually interpolate at a later stage. This principle is generally true if we exclude synthesis algorithms featuring stochastic components, and parameters with strong nonlinearities (e.g. binary switches). Generally, the opposite of this principle does not hold. Proximity in the timbre space does not necessarily imply similar parameter configuration. The TSAM itself can be used to verify these principles. A further reduction can be achieved limiting the individual range of interest of each parameter.

Given k variable synthesis parameter, the synthesis state space $\mathbf{I}$ (set of unique parameter permutations) is given by the Equations (1)-(3) [9].

$$\mathbf{I} = [\mathbf{i}_1, \mathbf{i}_2, \dots, \mathbf{i}_n] \tag{1}$$

$$\mathbf{i} = [i_1, i_2, \dots, i_k] \tag{2}$$

$$n = \prod_{j=1}^{k} \frac{\max(i_j) - \min(i_j)}{\text{step}(i_j)} \tag{3}$$

Each synthesis state is represented with a vector $\mathbf{i}$ with dimensionality k, as in Equation (2), while n, the number of vectors in $\mathbf{I}$, depends on the individual range and step size of the k parameters, as in (3). $\mathbf{I}$ is the synthesis state space we consider for the timbre analysis, presenting dimensionality k and cardinality n.

2.1 Descriptors Set and Computation

For each state $\mathbf{i}$ of the sound synthesizer we compute a set of audio descriptors, that we indicate with $\mathbf{d}$, representing the timbral descriptors of the resulting synthetic sound. A large set of low-level computational descriptors, including eventual redundancies, is essential for the detailed timbre analysis we require in this context. A few higher-level timbre descriptors (e.g. brightness, noisiness, coloration), often subjective and language dependent semantic [10], are suitable to discriminate sounds with major timbral differences, but in this context they fail to capture the subtle sonic nuances determined by small variations of the synthesis parameters.

A posterior descriptor selection is possible considering the quality metric we present in this paper. The method is independent of the specific descriptors set. In the TSAM we use the CUIDADO features set [11] implemented in the IRCAM descriptors object for Max/MSP. The set includes spectral and perceptual features listed in Table 1. It includes 24 scalar and 7 vectorial descriptors, as specified in the dimensionality column, resulting in a dimensionality q of $\mathbf{d}$ equal to 108, as in (4). Some of the scalar descriptors in the set are closely related to traditional timbre labels (e.g. spectral centroid to brightness).

$$\mathbf{d} = [d_1, d_2, \dots, d_q] \tag{4}$$

Descriptor Name	*Dimensionality*
Total Energy	1
Signal Zero Crossing Rate	1
Spectral Centroid	1
Spectral Crest	4
Spectral Decrease	1
Spectral Flatness	4
Spectral Kurtosis	1
Spectral Rolloff	1
Spectral Skewness	1
Spectral Slope	1
Spectral Spread	1
Spectral Variation	1
Perceptual Odd To Even Ratio	1
Perceptual Spectral Centroid	1
Perceptual Spectral Decrease	1
Perceptual Spectral Deviation	1
Perceptual Spectral Kurtosis	1
Perceptual Spectral Rolloff	1
Perceptual Spectral Skewness	1
Perceptual Spectral Slope	1
Perceptual Spectral Spread	1
Perceptual Spectral Variation	1
Perceptual Tristimulus	3
Sharpness	1
Spread	1
Noise Energy	1
Noisiness	1
Chroma	12
MFCC	13
Relative Specific Loudness	24
Perceptual Model	24

Table 1. List of descriptors used in the TSAM.

The descriptors listed above are computed on a short temporal window, typically in the range 2 ms to 200 ms. They provide an instantaneous sonic representation sufficient to characterize only absolutely periodic sounds. In synthesis states we may observe and hear low rate timbre

variations, spanning beyond the largest temporal window we consider for the descriptors. Hence an appropriate characterization of the timbre requires computation and merges of descriptors computed from multiple short time windows. We propose two analysis modes named 'sustain' and 'envelope' mode. In the first, given a synthesis state **i**, we compute *m* descriptor vectors and we combine these taking their mean and optionally their range, as in Equation (5), doubling the dimensionality of the descriptor set. The second approach simply concatenates the *m* descriptor vectors into a single vector, as in Equation (6), increasing the dimensionality by *m* times.

$$\mathbf{i} \leftrightarrow \mathbf{d} = \begin{bmatrix} \text{mean}(\mathbf{d}_1, \mathbf{d}_2, \dots, \mathbf{d}_m) \\ \max(\mathbf{d}_1, \dots, \mathbf{d}_m) - \min(\mathbf{d}_1, \dots, \mathbf{d}_m) \end{bmatrix} \quad (5)$$

$$\mathbf{i} \leftrightarrow \mathbf{d} = \begin{bmatrix} \mathbf{d}_1 \\ \vdots \\ \mathbf{d}_m \end{bmatrix} \quad (6)$$

Considering the synthesis as an binary process, and the sound generated as almost periodic, the first approach provides a sufficient approximation of the timbre. When the synthesis produces dynamic timbres, such as texture-like sounds, or when ADSR envelopes are applied to amplitude and other parameters, the second approach is preferred. However also in presence of ADSR envelopes, we can still use the first approach, analyzing only the sustain phase of the synthesis, intentionally discarding the attack, decay and release phases, or because these do not significantly change within the parameter space **I** we analyze.

The concatenation of short-term static descriptors to analyze timbral dynamics is a simplification with respect to the use of dynamic descriptors computed on longer temporal windows. However this approach reduces the time needed to execute the timbre analysis and allows users to change the merging mode from 'sustain' to 'envelope' and vice versa without repeating the analysis.

In the TSAM implementation, presented in Section 4, the computation of the descriptors is completely automated. Users are only required to identify the *k* variable parameters of the synthesizer, their range, step size, number of descriptor vectors per state m, and analysis mode. The tool computes **I** and drives the synthesizer with one **i** at a time, computing and storing m vectors **d**. For analysis in envelope mode, the tool also manages the triggering of the synthesizer at every **i**. Users can further specify the temporal unfolding of the analysis, selecting only a subset of the ADSR envelope. Advanced options related to the descriptor computation, such as window size, hop size, sampling rate, are exposed as well.

2.2 Descriptor Quality Metric

The quality metric we compute for each descriptor is aimed at capturing the four characteristics listed below.

- Noisiness: deviation of the descriptor from its mean given a synthesis state **i**.
- Variance: spread of descriptor value across the synthesis state space **I**.
- Independence: uniqueness of the descriptor variation pattern across the synthesis state space **I**.
- Correlation: coherence of the descriptor variation with synthesis parameters across the synthesis state space **I**.

Ideally, a descriptor representative of **I** should present low noisiness, high variance, high independence, and high correlation. High noisiness indicates that a particular descriptor and the associated timbral characteristic also varies when synthesis parameters are fixed, and therefore its eventual variance across **I** may be not significant. A descriptor with low variance reveals that the related timbral characteristic does not change significantly when varying the synthesis parameters. Descriptors varying with a similar trend are redundant, and thus less significant, when computing a dimensionality-reduced timbre space modeling **I**, instead those more independent carry a larger amount of information. Descriptors can also be highly independent when varying randomly across **I**. We address this by also including the correlation between descriptor and parameters in the metric, as we expect representative descriptors to change accordingly to one or more synthesis parameter.

For each descriptor, we compute the nosiness $\mathrm{N}_{x,\mathbf{i}}$ from the *m* descriptor vectors in synthesis state **i** before these are merged, as per Equations (5) and (6). The subscript *x* is the index identifying the descriptor across the set of *q* computed in the TSAM. For 'sustain' mode, we measure the deviation of the descriptor *x* in the state **i** using the Relative Mean absolute Difference (RMD), as in Equation (7). The RMD is a scale invariant measure of statistical dispersion, hence allows the comparison of heterogeneous descriptors. For 'envelope' mode, $\mathrm{N}_{x,\mathbf{i}}$ is estimated as the zero crossing rate, as in Equation (8), of the forward second order finite difference (discrete approximation of the second order derivative) of the series of *m* descriptors, as in (9). This represents the rate at which a descriptor inverts its trend (from increasing to decreasing and vice versa) in the analyzed envelope. Noisy descriptors invert their trends at higher rates.

$$\mathrm{N}_{x,\mathbf{i}} = \sum_{j=1}^{m} \sum_{k=1}^{m} \left| d_{x,j} - d_{x,k} \right| \Big/ \left| \sum_{j=1}^{m} d_{x,j} \right| (m-1) \quad (7)$$

$$\mathrm{N}_{x,\mathbf{i}} = \frac{1}{m-2} \sum_{j=2}^{m-1} \mathbb{I}\{\Delta^2(d_{x,j})\Delta^2(d_{x,j-1}) < 0\} \quad (8)$$

$$\Delta^2(d_{x,j}) = \sum_{k=0}^{2} \binom{2}{k} (-1)^{2-k} d_{x,j+k} \quad (9)$$

In Equations (7)-(9), $d_{x,j}$ represents the *x*-th descriptor in the set of *q*, from the *j*-th vector **d** out of the *m* computed for each state **i**. The indicator function $\mathbb{I}\{\ \}$ is equal to 1 if its argument is true, 0 otherwise. $\Delta^2(\ )$ is the forward second order finite difference function. The overall noisiness of each descriptor N_x is computed by taking the average over the set of synthesis unique states **I** we analyze.

Variance, independence, and correlation are computed across **I**, after the *m* descriptors are merged as in (5)-(6). The same method is used for both 'sustain' and 'envelope' modes. The variance V_x is computed as the RMD over the *n* synthesis states **i**. We use the same expression as in (7), replacing *m* with *n*, but in this case $d_{x,j}$ is the *x*-th descriptor in the set of *q*, from the *j*-th vector **d** out of the *n* we compute across **I**.

We assume that descriptors are independent if poorly correlated, therefore we compute I_x taking the complement of the averaged absolute value of the correlation coefficient between the descriptor *x* and the other *q-1* descriptors over **I**, as in Equation (10). Both positive and negative correlations indicate dependence, therefore we take the absolute value of the correlation coefficient corr( ). We subtract 1 from the summation to remove the correlation coefficient of the descriptor with itself, when *j=x*. Finally, the correlation C_x between descriptors and parameters is computed taking the average correlation coefficient between the *x*-th descriptor and the *k* variable synthesis parameter, as in Equation (11).

$$I_x = 1 - \frac{1}{q-1}\left[\left(\sum_{j=1}^{q} \left|\mathrm{corr}\left(\mathbf{d}_{x,\mathbf{I}}, \mathbf{d}_{j,\mathbf{I}}\right)\right|\right) - 1\right] \quad (10)$$

$$C_x = \frac{1}{k}\sum_{j=1}^{k} \left|\mathrm{corr}\left(\mathbf{d}_{x,\mathbf{I}}, \mathbf{i}_{j,\mathbf{I}}\right)\right| \quad (11)$$

In (10) and (11) with $\mathbf{d}_{x,\mathbf{I}}$ we represent the vector containing the *n* values of the *x*-th descriptor computed over the synthesis state space **I**, while $\mathbf{i}_{j,\mathbf{I}}$ represents the vector containing the *n* values of the *x*-th synthesis parameter over **I**. Note that according to (5) and (6) each descriptor may contribute with more than one component in each vector **d**. In particular, for 'sustain' mode we have two components per descriptor if the range is included in the analysis, whereas for the 'envelope' mode we have *m* components per descriptor. Therefore we compute multiple V_x, I_x and C_x per each *x*-th descriptor, and use their average in the quality metric we introduce next.

The quality metric S_x of each descriptor is computed from the individual noisiness, variance, independence, and correlation as in Equation (12). The noisiness, being an undesirable feature, lowers the value of S_x. The four components are combined using individual weights *w*.

$$S_x = w_V V_x + w_I I_x + w_C C_x - w_N N_x \quad (12)$$

The selection of the *w* values depends on the aim and context of the timbre analysis, and also on individual preferences. For instance, when analyzing a synthesizer configuration with a texture-like timbre, we expect considerable sonic variation within each synthesis state **i**, therefore the noisiness has no significance and w_N should be close to zero. If the purpose of the analysis is the sole study of the synthesizer timbre through the descriptors, their independence has little relevance. Instead when descriptors are used for mapping purposes, as in Section 3, the independence has a higher significance. In the TSAM, the default values of the weights are 0.33 for variance, independence and correlation, and 0.66 for noisiness. Users can change these in the unitary range. The four components of the quality metric have different ranges. I_x and C_x span between [0,1], while N_x and V_x can be zero but do not have a theoretical maximum. In the TSAM we include the option to normalize these to the unitary range, easing the balancing through individual weights. However when comparing the quality metrics S_x across different synthesizers, or between different state spaces **I** of the same synthesizer, normalization should not be used. In the TSAM we also rank also the *k* synthesis parameter by their average correlation with the *q* descriptors, computed as in (11) but replacing *k* with *q* and taking *x* as the summation index. Furthermore for each parameter, the TSAM displays the two descriptors with associated highest and lowest correlation, and vice versa.

3. TIMBRE SPACE MODELING AND MAPPING

Audio descriptors have been extensively used for visualization, measurement, classification, and recognition of sounds. Works proposing the timbre as a control structure for sound synthesis [12] or for interactive sonic systems have recently proliferated [13]–[24]. These allow for explicit control of psychoacoustic characteristics of the generated sound, hiding synthesis parameters from users, simplifying the user interaction, facilitating the search for specific timbres, and enhancing the expressivity of the system. Similar benefits are provided by synthesis methods using a timbre representation derived by a prior analysis stage of the target sound [25], [26]. A model relating parameters to sonic response of the sound synthesizer is necessary to implement explicit timbre control. Our generic approach, introduced in [7] and extended here, derives a model from the prior analysis stage, and therefore it is independent of the specific synthesis method and implementation.

The generative mapping is based on unsupervised machine learning techniques, and it provides a low dimensional and perceptually related synthesis control. The mapping maximizes the breadth of the explorable sonic space covered by the synthesis space **I**, and minimizes possible timbre losses due to the reduced dimensionality of the control space (i.e. few-to-many mapping). The timbre response analysis described in the previous section returns a synthesis space **I**, with dimensionality *k*, and a descriptor space **D**, with dimensionality *q*. Both spaces present *n* entries **i** and **d**, which are pairwise associated, representing a basic model relating parameters and timbre. Hence we can explicitly express a timbre through the *q* descriptors (e.g. mapped on a large bank of faders), find the closest entry in **D**, and drive of the synthesizer with the associated parameter set **i**. Such control is affected by several drawbacks: the high dimensionality of the timbre-based control, with *q* generally much greater than *k*; the lack of accuracy due to the large parameter step size we use in the analysis stage (3); entries in the timbre space **D**

are not evenly distributed as in **I**, hence regions of **D** with low density determine a poor system response.

The real dimensionality of **D** is usually much less than q. Generally the data of interest lies on an embedded non-linear manifold within the q-dimensional space. Therefore we reduce the dimensionality of **D**, using Isomap, down to two or three dimensions, which are easy to map to general-purpose controllers with low cognitive complexity. In the TSAM users can explore the application of 34 different dimensionality reduction methods [27].

Before reducing the dimensionality of **D**, we use the quality metric S_x to discard those descriptors with a low score. Particularly noisy or poorly correlated descriptors present a large variance that have a significant impact in the dimensionality reduction stage, but this would not be not representative of the parameter-to-timbre relationship, corrupting the timbre space mapping. The selection of descriptors based on the quality metric determines improvements in accuracy and usability against our previous approach. Alternatively, users can bypass the dimensionality reduction stage, and explicitly specify the two or three descriptors composing the low dimensional timbre space we use for the mapping to synthesis parameters.

To address the issue of the possible unresponsiveness of the timbre space due to arbitrary distribution in **D** we apply an iterative algorithm based on the Voronoi tessellation, derived from [28], that redistribute the n entries **d** into an uniformly distributed square or cube, while preserving the local neighborhood relationships (homomorphic transformation). The inverse of this transformation represent the required mapping to project a generic multidimensional control space **C** onto the case specific timbre space. Hence we use an Artificial Neural Network (ANN) to learn a function $m(\ \ )$ approximating the inverse of the redistribution process. We use $m(\ \ )$ to project the generic multidimensional control vector **c** onto the dimensionally reduced timbre space $\mathbf{D}^*$. The ANN includes a single hidden layer and therefore can be trained efficiently using a non-iterative algorithm [29]. In Figure 1 we show an example of a highly clustered timbre space reduced to three dimensions, and its transformation to a uniform cube. The side arrows identify the two stages of the mapping computation. In the TSAM we provide also an alternative mapping, skipping the ANN and computing the synthesis parameters directly from the uniformly distributed timbre space.

In the final stage of the mapping we compute the parameters to interact with the sound synthesizer. We use $\mathbf{d}^*$ to represent a descriptor vector in the dimensionality reduced timbre space $\mathbf{D}^*$. Driving the synthesis with the parameters **i** associated with the $\mathbf{d}^*$ closer to $m(\mathbf{c})$ may lead to discontinuities, that in turn may generate glitches in the sonic output. These are due to the coarse parameter step size used in the analysis stage, and due to the not one-to-one relationship between parameters and sound. Two synthesis states **i**, far apart in the synthesis state space **I**, may be associated identical or similar descriptor vectors **d**, hence close in **D**. The latter is an implicit drawback of any methods for controlling sound synthesis from any representation of the generated signal.

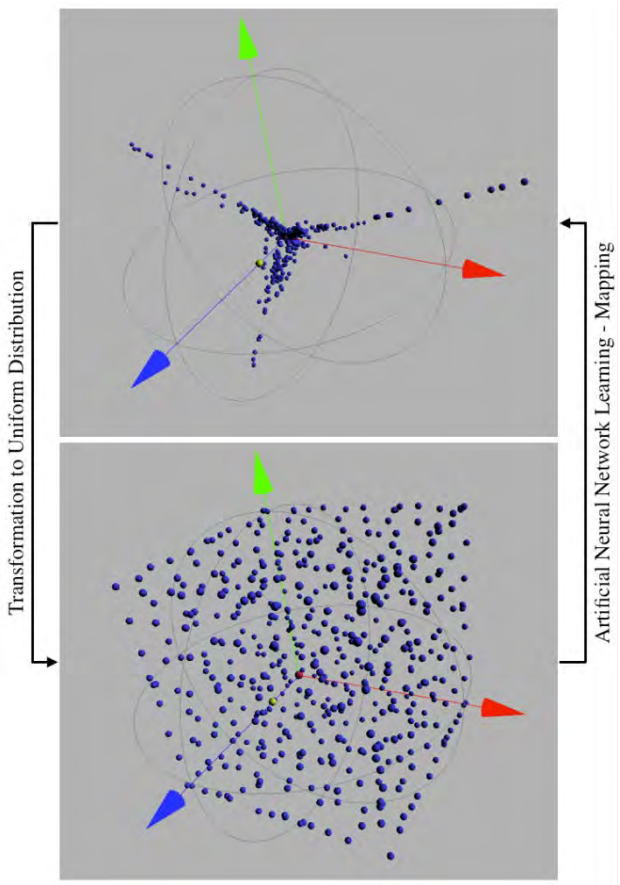


Figure 1. Example of a timbre space reduced to three dimensions, and related transformation to a uniform cube.

We address these issues computing the synthesis parameter by spatial interpolation, including only entries of $\mathbf{D}^*$ from the neighborhood the current state **i**. The set of parameters driving the synthesizer $\mathbf{i}_{ctrl}$ is computed by Inverse Distance Weighting (IDW) as in Equations (12) and (13), where $\|\ \ \|$ represent the Euclidean distance.

$$\mathbf{i}_{ctrl} = \frac{\sum_{j=1}^{N} \mathbf{q}_j(m(\mathbf{c})) \cdot \mathbf{i}_j}{\sum_{j=1}^{N} \mathbf{q}_j(m(\mathbf{c}))} \tag{12}$$

$$\mathbf{q}_j(m(\mathbf{c})) = \frac{1}{\left\| m(\mathbf{c}) - \mathbf{d}_j^* \right\|^p} \tag{13}$$

In (12) and (13) N represents the total number of points considered in the interpolation, and the $\mathbf{i}_j$ in (12) are those pairwise associated with the $\mathbf{d}_j^*$ in (13). In the TSAM instead of using the N closest point $\mathbf{d}_j^*$ in $\mathbf{D}^*$, we select those $\mathbf{d}_j^*$ that limit the maximum variation of $\mathbf{i}_{ctrl}$ between two consecutive iterations, that is the set of $\mathbf{d}_j^*$ associated with the $\mathbf{i}_j$ close to the current $\mathbf{i}_{ctrl}$ (within a user-defined distance). In Figure 2 we show an example of this interpolation points selection, where the green entries are the $\mathbf{d}_j^*$ related to $\mathbf{i}_j$ close to the current $\mathbf{i}_{ctrl}$, which is in turn associated with the yellow one in figure.

The set of $\mathbf{d}_j^*$ used for IDW interpolation may include entries distant from $m(\mathbf{c})$, but these will poorly contribute in (12). In the IDW, p represents the power parameter, which determines the influence of each point based on the distance. This value should be larger than the dimensionality of the reduced timbre space $\mathbf{D}^*$, and increasing p closer points has larger weight. In the TSAM, the $\mathbf{i}_{ctrl}$ maximum instantaneous distance and interpolation power parameter p, are among the options exposed to users to tune in real time the timbre mapping response. The TSAM provides interactive timbre space visualizations, such as those in Figure 1 and 2.

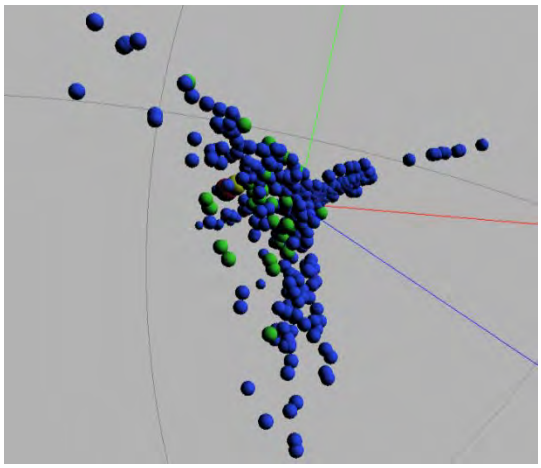


Figure 2. Detail of a timbre space reduced to three dimensions. The green entries are those used in the interpolation to compute the synthesis parameter, because close to the yellow current entry in the synthesis state space.

4. IMPLEMENTATION AND USAGE

The TSAM[1] is an open-source software implemented in in Max/MSP using FTM extension[2] [30], supported by a background engine written and compiled in MATLAB. The analysis of the synthesis timbre, the real-time timbre space mapping and the visualizations are computed in Max/MSP, whereas the background engine computes the descriptor quality and the timbre space mapping (dimensionality reduction, redistribution, ANN training), taking as input the outcome of the analysis stage. The two components of the system communicate via Open Sound Control (OSC) protocol and large matrices are exchanged using files. The TSAM can host software synthesizer developed using Steinberg's Virtual Studio Technology (VST). It acts as a wrapper for VST synth, providing a fully integrated environment. The TSAM allows full control of all parameters for analysis and mapping purposes. It captures the synthetized signal for descriptor computation and playback, and manages the global state of the synthesizer when saving and restoring presets. In Figure 3 there is a screenshot of the main TSAM GUI. This exposes a large number of options for further exploration of the mapping method we propose, and also for customizing analysis, mapping computation, real-time control, and visualization. Default settings are provided for basic use. Users can load a VST synth and select up to 10 variable parameters, their range, analysis step size, and the number of vectors m per state $\mathbf{i}$. Advanced analysis options include digital signal processing settings and analysis timing with respect to the synthesis triggering (note-on and note-off messages). The TSAM estimates and shows the total analysis time, and users may opt to reduce the parameter step sizes, in (3), when this is excessive. Thereafter the analysis is carried out automatically. In Section 2 we discussed two analysis modes, 'sustain' and 'envelope' respectively. These, besides the automatic mode, can also be carried out manually. Users arbitrarily tune the synthesizer to a specific state $\mathbf{i}$, and request for the descriptor analysis of the related sonic response (both modes are supported). Furthermore we included the interactive 'sustain' analysis mode [7] where descriptor vectors $\mathbf{d}$ are computed while users vary in the MIDI mapped synthesis parameters in real-time, dynamically generating a stream of $\mathbf{i}$. The latter analysis mode does not guarantee to observe an identical number of descriptor vectors $\mathbf{d}$ per state $\mathbf{i}$, hence the noisiness in the quality metric result may be inconsistent.

When the analysis stage is completed, users can request the computation of the descriptor quality metric, which is visualized in the TSAM as shown in Figure 4. In the descriptors page, users can also specify the weights of Equation (12), enable the normalization of its components, find and rank the descriptors by highest score, observe the synthesis parameter ranking, and find the highest and lowest correlation between each parameter and descriptor. Furthermore, users can specify which subset of the 108 descriptors will be used for mapping purposes.

Options for the timbre space mapping computation include the dimensionality of the map, selection of the dimensionality reduction technique and the ANN activation function. The mapping can be tuned at runtime using the settings discussed in Section 3. The timbre analysis, quality metric, and mapping are saved into files that can be individually recalled through the TSAM presets.

5. DISCUSSION AND FUTURE WORK

We presented a generic tool that integrates functionalities to study and map the timbre of sound synthesizers. Preliminary studies demonstrated that the adoption of large sets of descriptors, and their selection based on the novel quality metric, improves the accuracy of the timbre-based interaction. The TSAM can be used for the study of the sonic response of synthesizers, for an explicit control of timbral character, or for a reduction of the synthesis control space, exposing only a few perceptually relevant control dimensions. Previous user studies on a system with a similar mapping approach demonstrated that synthesis parameters become transparent to users [31], which are exclusively focused on the timbral interaction. Future works include user studies with the TSAM to evaluate the

[1] http://stefanofasciani.com/tsam.html
[2] http://ftm.ircam.fr/

effectiveness of the timbre-based mapping, comparing it against traditional and alternative approaches to sound synthesis interaction, in performing and sound design scenarios. Moreover we will investigate the relevance of different descriptor categories for a more perceptually related sonic control.

Figure 3. TSAM main page, including options for analysis, mapping computation, real-time control, and visualization.

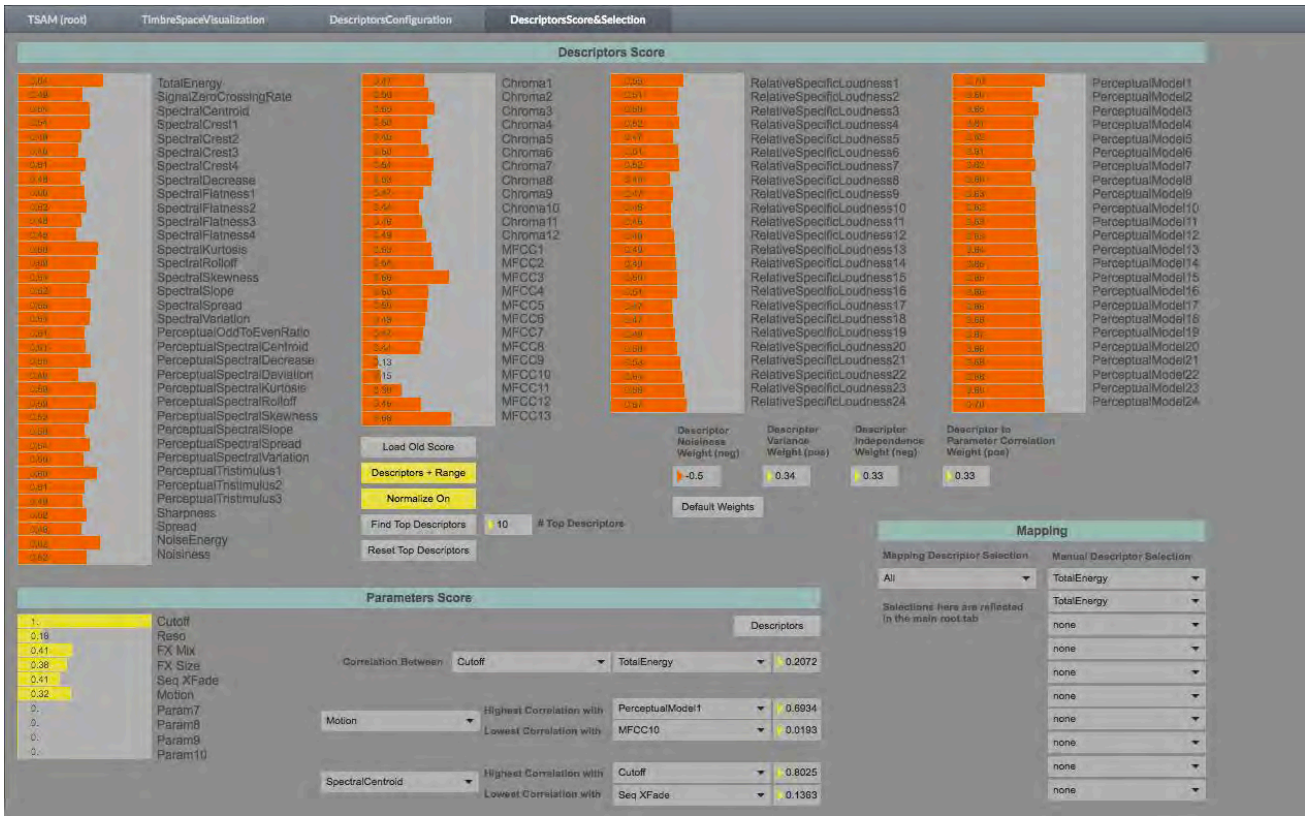


Figure 4. TSAM descriptor page, providing an insight into the timbre response and parameter relationship of the synth.

6. REFERENCES

[1] T. Wishart, *On Sonic Art*. Harwood Academic Publishers, 1996.

[2] S. McAdams and A. Bergman, "Hearing musical streams," *Comput. Music J.*, vol. 3, no. 4, pp. 26–43, 60, 1979.

[3] J. C. Risset and D. Wessel, "Exploration of timbre by analysis and synthesis," *Psychol. Music*, pp. 113–169, 1999.

[4] S. Fels, "Intimacy and embodiment: implications for art and technology," in *Proc. of the 2000 ACM workshops on Multimedia*, 2000, pp. 13–16.

[5] E. R. Miranda and M. M. Wanderley, *New digital musical instruments: control and interaction beyond the keyboard*. A-R Editions, Inc., 2006.

[6] D. Arfib, J. M. Couturier, L. Kessous, and V. Verfaille, "Strategies of mapping between gesture data and synthesis model parameters using perceptual spaces," *Organ. Sound*, vol. 7, no. 2, pp. 127–144, Aug. 2002.

[7] S. Fasciani, "Interactive Computation of Timbre Spaces for Sound Synthesis Control," in *Proc. of the 2nd Int. Symposium on Sound and Interactivity*, Singapore, 2015.

[8] W. A. Sethares, *Tuning, Timbre, Spectrum, Scale*. Springer Science & Business Media, 2005.

[9] S. Fasciani and L. Wyse, "Adapting general purpose interfaces to synthesis engines using unsupervised dimensionality reduction techniques and inverse mapping from features to parameters," in *Proc. of the 2012 Int. Computer Music Conf.*, Ljubljana, Slovenia, 2012.

[10] A. Zacharakis, K. Pastiadis, and J. D. Reiss, "An Interlanguage Unification of Musical Timbre," *Music Percept. Interdiscip. J.*, vol. 32, no. 4, pp. 394–412, Apr. 2015.

[11] G. Peeters, "A Large Set of Audio Features for Sound Description (Similarity and Classification) in the Cuidado Project," IRCAM, 2004.

[12] D. Wessel, "Timbre space as a musical control structure," *Comput. Music J.*, vol. 3, no. 2, pp. 45–52, 1979.

[13] A. Lazier and P. R. Cook, "Mosievius: feature driven interactive audio mosaicing," in *Proc. of the 7th Int. Conf. on Digital Audio Effects*, Napoli, Italy, 2003.

[14] M. Puckette, "Low-dimensional parameter mapping using spectral envelopes," in *Proc. of the 2004 Int. Computer Music Conf.*, Miami, US, 2004.

[15] C. Nicol, S. A. Brewster, and P. D. Gray, "Designing Sound: Towards a System for Designing Audio Interfaces using Timbre Spaces.," in *Proc. of the 10th Int. Conf. on Auditory Display*, Sydney, Australia, 2004.

[16] D. Schwarz, G. Beller, B. Verbrugghe, and S. Britton, "Real-time corpus-based concatenative synthesis with CARART," in *Proc. of the 9th Int. Conf. on Digital Audio Effects*, Montreal, Canada, 2006, pp. 279–282.

[17] M. Hoffman and P. R. Cook, "Feature-based synthesis: Mapping acoustic and perceptual features onto synthesis parameters," in *Proc. of the 2006 Int. Computer Music Conf.*, New Orleans, US, 2006.

[18] N. Schnell, M. A. S. Cifuentes, and J. P. Lambert, "First steps in relaxed real-time typo-morphological audio analysis/synthesis," in *Proceeding of the 7th Sound and Music Computing Int. Conf.*, Barcelona, Spain, 2010.

[19] T. Grill, "Constructing high-level perceptual audio descriptors for textural sounds," in *Proc. of the 9th Sound and Music Computing Int. Conf.*, Copenhagen, Denmark, 2012.

[20] A. Seago, "A New Interaction Strategy for Musical Timbre Design," in *Music and Human-Computer Interaction*, S. Holland, K. Wilkie, P. Mulholland, and A. Seago, Eds. Springer, 2013, pp. 153–169.

[21] A. Pošćić and G. Kreković, "Controlling a sound synthesizer using timbral attributes," in *Proc. of the 10th Sound and Music Computing Int. Conf.*, Stockholm, Sweden, 2013.

[22] N. Klügel, T. Becker, and G. Groh, "Designing Sound Collaboratively Perceptually Motivated Audio Synthesis," in *Proc. of the 14th Int. Conf. on New Interfaces for Musical Expression*, London, United Kingdom, 2014, pp. 327–330.

[23] S. Ferguson, "Using Audio Feature Extraction for Interactive Feature-Based Sonification of Sound," in *Proc. of the 21st Int. Conf. on Auditory Display (ICAD 2015)*, Graz, Austria, 2015.

[24] S. Stasis, R. Stables, and J. Hockman, "A Model For Adaptive Reduced-Dimensionality Equalisation," in *Proc. of the 18th Int. Conf. on Digital Audio Effects (DAFx-15)*, Trondheim, Norway, 2015.

[25] X. Serra and J. Smith, "Spectral Modeling Synthesis: A Sound Analysis/Synthesis System Based on a Deterministic Plus Stochastic Decomposition," *Comput. Music J.*, vol. 14, no. 4, pp. 12–24, 1990.

[26] T. Jehan and B. Schoner, "An audio-driven perceptually meaningful timbre synthesizer," in *Proc. of the 2001 Int. Computer Music Conf.*, Havana, Cuba, 2001.

[27] L. J. P. Van Der Maaten, E. O. Postma, and H. J. Van Den Herik, "Dimensionality reduction: a comparative review," Tilburg University Technical Report, 2009.

[28] H. Nguyen, J. Burkardt, M. Gunzburger, L. Ju, and Y. Saka, "Constrained CVT meshes and a comparison of triangular mesh generators," *Comput. Geom.*, vol. 42, no. 1, pp. 1–19, Jan. 2009.

[29] G. B. Huang, Q. Y. Zhu, and C. K. Siew, "Extreme learning machine: Theory and applications," *Neurocomputing*, vol. 70, no. 1–3, pp. 489–501, Dec. 2006.

[30] N. Schnell, R. Borghesi, D. Schwarz, F. Bevilacqua, and R. Muller, "FTM - Complex Data Structure for Max," in *Proc. of the 2005 Int. Computer Music Conf.*, Barcelona, Spain, 2005.

[31] S. Fasciani, "Voice-controlled interface for digital musical instruments," Ph.D. Thesis, National University of Singapore, Singapore, 2014.

PRECISION FINGER PRESSING FORCE SENSING IN THE PIANIST-PIANO INTERACTION

M. Flückiger, T. Grosshauser, G. Tröster
ETH Zurich, Electronics Laboratory
mfluecki@ethz.ch, grotobia@ethz.ch, gerhartr@ethz.ch

ABSTRACT

Playing style, technique and touch quality are essential for musical expression in piano playing. From a mechanical point of view, this is mainly influenced by finger pressing force, finger position and finger contact area size. To measure these quantities, we introduce and evaluate a new sensor setup suited for the in-depth investigation of the pianist-piano interaction. A strain gauge based load cell is installed inside a piano key to measure finger pressing force via deflection. Several prototypes of the finger pressing force sensor have been tested and the final sensor measures from $0\,\mathrm{N}$ to $40\,\mathrm{N}$ with a resolution smaller than $8\,\mathrm{mN}$ and a sample rate of $1000\,\mathrm{Hz}$. Besides an overview of relevant findings from psychophysics research, two pilot experiments with a single key piano action model are presented to explore the capability of the force sensor and to discuss possible applications.

1. INTRODUCTION

The piano action connects the key and the hammer motion by a combination of mechanical levers. Due to the repetition mechanism the key loses its mechanical connection to the hammer shortly before key bottom contact. Per key, the piano action can be modeled essentially as a one degree of freedom system, e.g., as explained by Smith and Van Duyne [1]. This is the main reason why digital and hybrid piano instruments usually measure the key velocity parameter defined by the MIDI standard.

A pianist, while playing, relies on emotional state, kinesthetic memory, intended image, experience, motor control and feedback of the instrument to achieve a certain musical expression during a performance. There are different playing styles, techniques and interpretations of emotional expressions, that lead to a complex and unique interaction at the interface of the instrument: the keyboard and the pedals. At the keyboard the pianist's input mainly translates into finger pressing force, finger position and finger contact area size.

Regarding the interaction with the key, Goebl et al. [2] reported that pianists have a large inventory of different key press actions and techniques to achieve fine timbral nuances. Further, it was mentioned by Tiedemann and Drescher [3] that expert pianists sometimes also use a unnecessary high pressure in quiet passages to achieve a certain musical expression. Independent of whether this quality of touch is audible in the generated sound, it appears to be of essential importance for the pianist.

We expect that this quality of the pianist-piano interaction is pronounced most in the finger pressing force. Therefore, this paper presents a new approach to measure finger pressing force with high precision for the in-depth investigation of the pianist-piano interaction.

The paper is structured as follows: Our sensor setup to measure the pianist-piano interaction is introduced, before the requirements for a finger pressing force sensor are discussed and the proposed design is explained and evaluated. Finally, the capability of the sensor is explored with two pilot experiments, where the piano key with the force sensor is installed in a single key piano action model.

1.1 State of the Art

Moog [4] was the first to present a keyboard capable of sensing finger position and after-touch force to augment musical expression possibilities of synthesizers.

Parlitz et al. [5] compared force measurements of amateur and professional piano players performing so-called tied finger exercises. For after-touch finger force sensing flexible printed circuit board (PCB) sensors with a resolution of $2\,\mathrm{N}$ at a sample rate of $80\,\mathrm{Hz}$ were installed on the key bed of a piano.

Grosshauser and Tröster [6] presented a solution to measure finger pressing force and finger position by using a flexible PCB with a force sensitive resistor based sensor matrix. Position and force resolution was 5 x 5 mm and $0.4\,\mathrm{N}$, respectively. Data was sampled at $100\,\mathrm{Hz}$. The participants of the study reported that the modification to the touch and feel of the key was perceivable but not distracting.

McPherson [7] followed another approach for the investigation of key touch features and presented a portable optical measurement system for continuous key motion sensing in piano playing. McPherson [8] also presented PCB-based touchpads that are installed on the piano key playing surfaces to capture finger position and finger touch area size by capacitive sensing with a sampling rate of $125\,\mathrm{Hz}$ and a resolution around $0.1\,\mathrm{mm}$.

Kinoshita et al. [9] installed a strain gauge based force sensor on a fixed position of a single key mounted in an upright piano to study the relation of finger pressing force

and the generated sound pressure level. The sensor had a measurement range from 0 N to 100 N, a resolution of 20 mN and data was sampled at 900 Hz.

Tiedemann and Drescher [3] designed a strain gauge based folded aluminum beam piano key for finger pressing force sensing on the entire playing surface of the key. Several of such keys were installed in a grand piano for force and efficiency monitoring. The sensor concept seems promising, but unfortunately a technical evaluation of the sensor is missing.

Finally, we want to mention the contribution of Askenfelt and Jansson [10–12]. In addition to the in-depth analysis of the mechanics of the piano action, also cues about finger pressing force levels were reported, although they do not mention the setup that was used for finger pressing force sensing.

2. TECHNICAL DESCRIPTION OF THE SENSOR SETUP

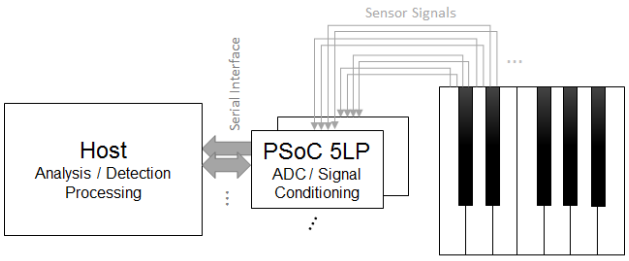


Figure 1. Our sensor setup to measure the pianist-piano interaction. The sensors are integrated into the keyboard and connected to Cypress PSoC 5LP microcontrollers that convert and condition the sensor signals. The digital data is captured and processed by the host. Host and microcontrollers communicate by a serial interface. The PSoC platform can also be used to communicate directly with MIDI or OSC commands.

The envisioned system design to measure finger pressing force, finger position and finger contact area size by integrating sensors unobtrusively into the keys of a piano is shown in Fig. 1. The system consists of a keyboard with sensors that are connected to Cypress PSoC 5LP [1] microcontrollers. These convert the analog sensor signals into digital streams that are sent to a central host for capturing and processing. Alternatively, the PSoC 5LP platform also offers OSC and MIDI connectivity, which can be used in combination with the sensors to provide new musical expression possibilities through standard interfaces. The solution is scalable by design and can be used for several piano keys or even a complete keyboard.

In this paper, we only focus on the finger pressing force measured with a load cell integrated in a single piano key, therefore we concentrate on a subset of this system. The digital data is sent via the serial interface to a host computer to record and analyze the data.

[1] http://www.cypress.com/products/32-bit-arm-cortex-m3-psoc-5lp

2.1 Perception Thresholds & Sensor Specification

For the in-depth investigation of the pianist-piano interaction, disturbances to the touch and feel of the piano keys should be kept to a minimum and unobtrusiveness of the design must be evaluated. Further, the following findings from literature have to be reflected in a finger pressing force sensor design.

King et al. [13] investigated perceptual thresholds for single and multi-finger haptic interaction and found fingertip force detection levels around 30 mN. Multi-finger interaction has no considerable effect on these single finger thresholds. Pang et al. [14] studied force detection thresholds by letting subjects squeeze two plates, grasped by the thumb and the index finger. One plate was fixed and one was movable but resisted with a constant, controlled force. The just-noticable difference they found was around 7 %, independent of the reference value of the resistive force ranging from 2.5 N to 10 N. Askenfelt and Jansson [11,12] observed finger pressing peak force levels of 8 N, 15 N and up to 50 N for pianists playing staccato at piano, at mezzoforte and at fortissimo level, respectively. Considerably lower levels were found for legato playing.

Engel et al. [15] showed that the timing of key presses by pianists varies in the same order of magnitude as the perception threshold for experienced listeners to detect asynchrony between two tone onsets, which is around 10 ms.

To sum up and by including a fair margin, an ideal piano key finger pressing force sensor should have a resolution around 10 mN from 0 N to 1 N and resolution relative to a few percent of the corresponding force level in the range from 1 N to 50 N. Data should be sampled at an interval of 1 ms.

2.2 Load Cell

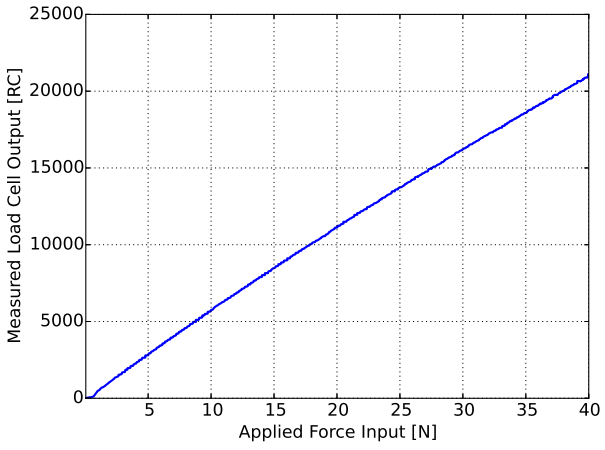


Figure 2. Measured input-output characteristic of the load cell. RC refers to raw count. Resolution is smaller than 2 mN per raw count. Input force was applied with the compression test machine shown in Fig. 5.

There are different force and pressure sensors based on diverse principles available on the market. We decided to use

a Honeywell FSS1500NS [2] strain gauge based load cell, because of its miniature size, weight and satisfying performance in earlier projects [16, 17].

Fig. 2 shows the measured input-output characteristic of the load cell for a force input range from $0\,\mathrm{N}$ to $40\,\mathrm{N}$. Using a 16 bit ADC, where one bit of precision is lost because of the differential load cell output, the resolution is smaller than $2\,\mathrm{mN}$. The load cell outputs a constant offset, if no force is applied. This offset is set to zero in the measurements presented in this paper. The input force was applied with a test machine manufactured by Zwick [3] . The procedure is explained at the end of the next subsection.

2.3 Integration into the Piano Key

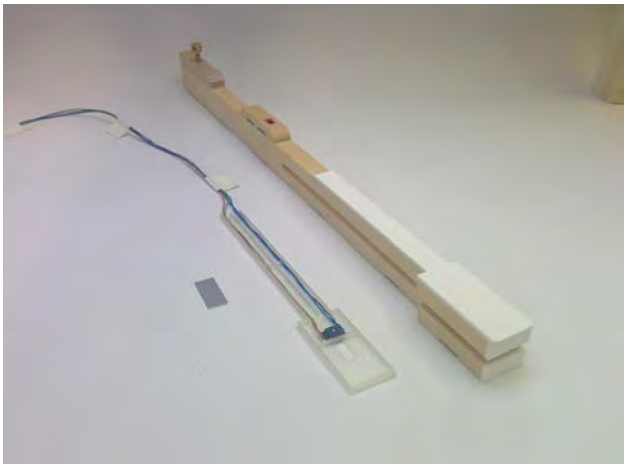


Figure 3. From left to right: small stainless steel plate, 3D printed support with installed load cell and piano key with the cut. Before the cut, the key weighted $78.6\,\mathrm{g}$. After the cut the key had a weight of $71.3\,\mathrm{g}$ and all the other components together scaled $6.6\,\mathrm{g}$. This sums up to a change in weight of less than $1\,\%$.

Several sensor prototypes to measure finger pressing force in piano playing have been evaluated to optimize measurement range, sensitivity, linearity, deflection and force leakage, unobtrusiveness and mechanical stability.

The most promising concept is presented hereafter and is applicable to both, white and black piano keys. The sensor consists of four components: the wooden piano key with a cut, a 3D printed support to hold and position the load cell and a small steel plate. A picture of the components is shown in Fig. 3. The load cell is mounted on the support, before the support is aligned and glued on the first cut surface of the piano key. The steel plate serves as the counterpart of the load cell tip and is glued on the second cut surface. The steel plate and the load cell tip keep steady contact and the steel plate prevents the tip from notching the wood.

A sketch of the piano key sensor cross section is shown in Fig. 4. Due to the cut, the major part of the finger pressing

[2] http://sensing.honeywell.com/products/force-sensors

[3] http://www.zwick.com/en/products/static-materials-testing-machines/testing-machines-up-to-5-kn/zwicki-line-testing-machines.html

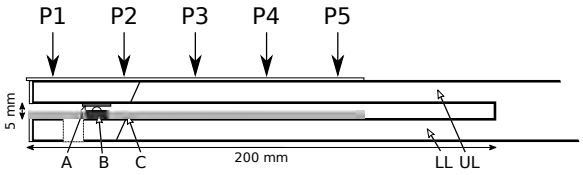


Figure 4. Piano key sensor cross section. Component arrangement, working principle and evaluated pressing positions P1 to P5. A: steel plate, B: load cell, C: 3D printed support, LL: lower leg, UL: upper leg.

force is deflected through the load cell, the minor part is leaked through the remaining wooden connection of the key. This leakage depends on cut position, cut length and cut height, and varies with finger pressing position along the endwise axis of the key. The cut height measures $5\,\mathrm{mm}$ and is given by the dimensions of the load cell, the steel plate and the 3D printed support. The length of the lever is the limit of the cut length, but the length also influences the mechanical stability of the key. The chosen cut has a length of $200\,\mathrm{mm}$ and was made at the center of the key height, due to symmetry and stability reasons. Prototypes with a cut closer to the key touching surface suffered from mechanical instability.

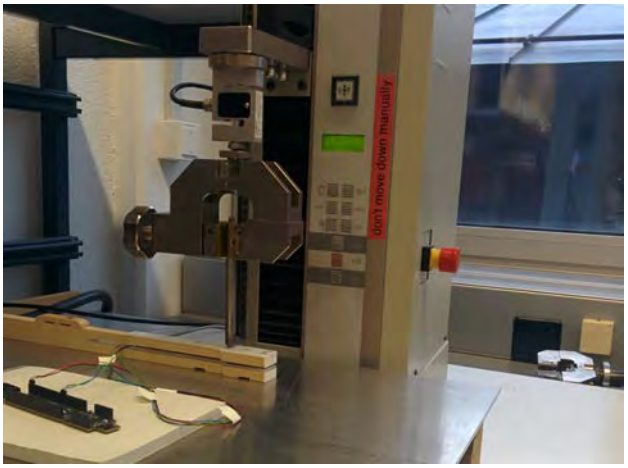


Figure 5. Finger pressing force sensor evaluation with the compression test machine. The steel peg has a diameter of $10\,\mathrm{mm}$ and a spherical tip. In this picture, the tip applies a controlled force to position P2 of the key.

The usage of up to three load cells in a row or with a triangulation approach was evaluated as well. However, the precise alignment of multiple load cells proved to be difficult, although using more load cells increases measurement range and decreases force leakage. Further, there is a limitation in the precision of the cut in wood working. To circumvent errors because of misalignment we decided to use a single load cell instead. The load cell position is a trade off between mechanical stability, measurement range and the amount of force leakage.

The sensor characteristic was evaluated by putting the key flat on a measurement table. The Zwick test machine was configured for controlled pressing with a steel peg, as

shown in Fig. 5. The force was applied to five equally spaced positions on the piano key sensor. These pressing positions are highlighted in Fig. 4.

The result of this measurement is presented in Fig. 6. The finger pressing force sensor preserves the linear characteristic of the load cell, but shows a position dependency along the endwise axis of the key. For position P1, force input is amplified due to the lever connection to the load cell and this far most position from the remaining wooden connection has minimum force leakage. For the following positions, P2 to P5, the input force deflected by the load cell decreases, while the force leakage increases. In consequence, also the resolution of the sensor is position dependent. Position P5 has the lowest resolution around $8\,\mathrm{mN}$ but remains below the resolution requirement of $10\,\mathrm{mN}$ from Subsection 2.1.

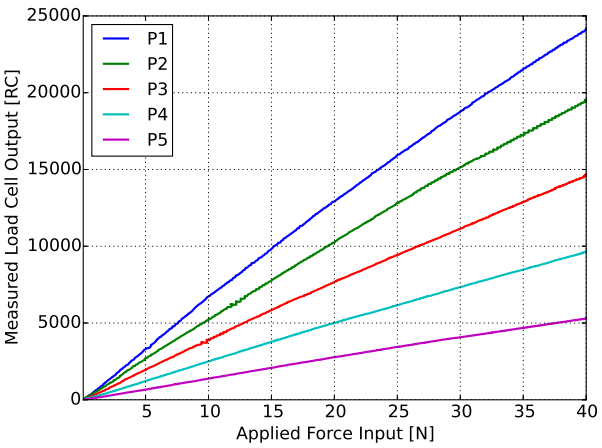


Figure 6. Measured input-output characteristic of the finger pressing force sensor for input force applied to five different positions along the endwise axis of the key. RC refers to raw count. The corresponding pressing positions on the key are highlighted in Fig. 4. Input force was applied with the compression test machine shown in Fig. 5.

The position dependency of the presented finger pressing force sensor needs to be compensated for realistic pianist-piano interaction measurements. As finger pressing position is one of the important interaction parameters anyway, we are working towards a solution, which is not part of this publication.

However, a finger position sensor like the one presented by McPherson [8] would be suited to compensate such a position dependency along the endwise axis of the key. For compensation, a gain for each position of the data presented in Fig. 6 can be calculated and the actual gain for a measured position can be estimated by linear interpolation. A simulation with this approach and the presented data showed that a position resolution of $0.5\,\mathrm{mm}$ along the endwise axis of the key keeps the maximum relative force error below $1.5\,\%$. Sensitivity to errors is highest at position P5, because it requires the largest gain factor for compensation.

3. SINGLE KEY PIANO ACTION MODEL

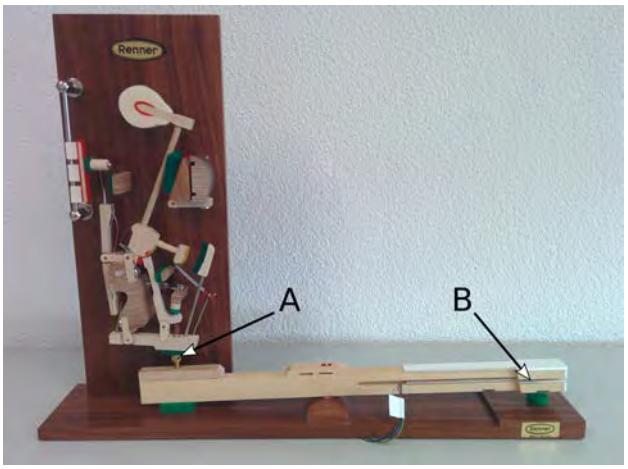


Figure 7. The piano key with the force sensor installed in the single key upright piano action model. The action model has the same mechanics found in complete upright piano actions. The major difference is that the hammer hits a metal bar instead of a string. A: connection of the key to the hammer action, B: force sensor.

In the previous section we showed that the integration into the key preserves the characteristic of the load cell but the position dependency needs to be compensated with a finger position sensor.

To assess the dynamic performance of the force sensor inside the piano key, it was installed in a single key upright piano action model as shown in Fig. 7. This action model has the same mechanics as found in a complete upright piano with the major difference that a metal bar is hit by the hammer instead of a string. The compression test setup of Fig. 5 that was used for the static assessment could not be used for a precise dynamic evaluation, because of its limits in maximum velocity and sampling frequency [4].

Hence, we decided to evaluate the force sensor in the piano action model by applying balance weights to the key, before the capability of the sensor is explored with two pilot experiments.

3.1 Evaluation of the Sensor in the Piano Action Model

The cut we made into the piano key generated two wooden legs of equal thickness: the upper leg (UL) and the lower leg (LL), see Fig. 4. The load cell senses in between these two wooden legs. Besides the desired sensitivity to finger pressing force, also modulations of the sensor gap height have an influence on the load cell output. Furthermore, the two legs have slightly different mechanical properties. The upper leg is covered with the white plastic touch surface of the key, which increases stiffness. The lower leg carries $6.6\,\mathrm{g}$ of additional weight from the load cell, support and cables.

[4] Maximum velocity is limited to $0.025\,\frac{\mathrm{m}}{\mathrm{s}}$ in contrast to typical key pressing velocities of $0.1\,\frac{\mathrm{m}}{\mathrm{s}}$ to $0.6\,\frac{\mathrm{m}}{\mathrm{s}}$ [11]. The applied force is sampled at $12.5\,\mathrm{Hz}$. Position P5 has a key displacement range of $\approx 3\,\mathrm{mm}$. This yields 1.5 force samples to investigate a key press.

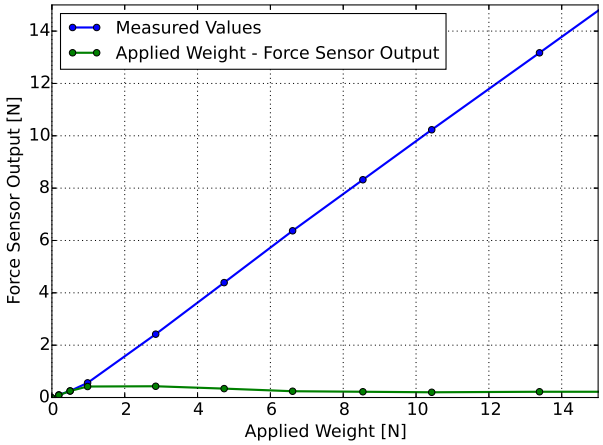


Figure 8. Evaluation of the force sensor inside the piano key in the piano action model. The dots mark the measured force sensor output of nine balance weights that were applied at the midpoint of position P1 and P2of the key. It takes $\approx 1\,\mathrm{N}$ to press down the key and launch the hammer.

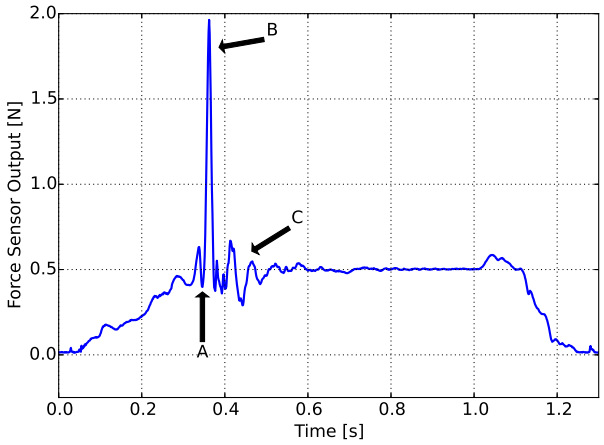


Figure 9. Sensor response to a $100\,\mathrm{g}$ balance weight that was applied to the piano key installed in the action model. A: escapement of the repetition mechanism, launching the hammer $16\,\mathrm{ms}$ before B: key-bottom contact, C: damped oscillation of the hammer after strike.

This leads to a smaller offset of the sensor output in the case, where the piano key is mounted in the action model compared to the case, where it is lying flat on a table. Further, Fig. 8 shows the influence on the response of the force sensor installed inside the piano key of the action model, when different balance weights are applied to the piano key. Thus, the weight applied to the key and the load from the hammer action on the key lead to a small but measurable modification of the sensor gap height.

To give an indication of the order of magnitude, a change of the force input from $0\,\mathrm{N}$ to $1\,\mathrm{N}$ applied to position P2 of the piano key under test in Fig. 5, corresponds to a compression around $50\,\mu\mathrm{m}$ of the piano key and sensor.

The force signals presented in the following two pilot experiments are not compensated with regard to this effect and finger pressing position was fixed at the midpoint of position P1 and P2.

3.2 Dynamic Response to a Balance Weight

Fig. 9 shows the response of the sensor to a $100\,\mathrm{g}$ balance weight that was applied to depress the piano key installed in the action model. After the application of the weight, the key is accelerated towards the key bed. The dip in force level, occurring $16\,\mathrm{ms}$ before the maximum peak force level, marks the escapement of the hammer repetition mechanism. The maximum force peak is generated by the deceleration of the key colliding with the key bed, usually referred as key-bottom contact [18]. Meanwhile, the hammer strikes the metal bar and falls back for potential relaunch. This mechanic action is fed back to the key and manifests as a damped oscillation in the signal of the force sensor. In addition, a structure-borne vibration, created by the hit of the metal bar by the hammer, superimposes this mechanical action feedback. Thereafter the signal reaches a steady state value around $0.5\,\mathrm{N}$, although $1\,\mathrm{N}$ was applied. This deviation was expected from the measurement presented in Fig. 8.

3.3 Measurement of a Finger Sequence

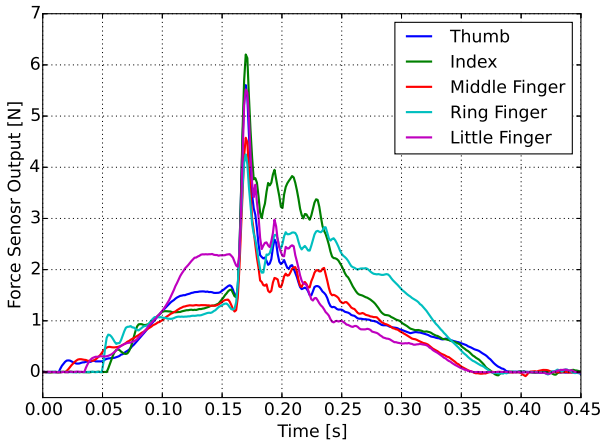


Figure 10. Overlay of five short notes played by a pianist at piano level. The notes were played with struck touch and the finger that presses the key was changed for every struck. The individual key strucks of the sensor signal were cut and realigned for comparison.

Fig. 10 shows the result of the second pilot experiment. A pianist played five short notes at piano dynamic level in a continuous recording. The notes were played with the right hand and struck touch. The finger pressing the key was changed from the thumb, to the index, to the middle finger, to the ring finger and finally to the little finger. For all fingers the envelope of the force signal shows the characteristic shape of a struck touch, cf. [9,19]. Differences in peak level, in playing efficiency [5] and in the shape of the force signal are apparent, although all notes were aimed to be played at the same dynamic level and with the same duration.

[5] A possible definition of playing efficiency can be found in [3].

4. CONCLUSION

We presented a new sensor concept to measure finger pressing force in the pianist-piano interaction. The sensor is capable to sense subtle changes of touch in piano playing but demands a finger pressing position measurement for compensation.

Further, we showed that for high precision sensing inside piano keys, special care has to be taken to account for disturbances due to material properties. This holds in general for traditional classical musical instruments, where wood is often encountered as a material. In addition, also the disparity of different piano models and tolerances in manufacturing have to be considered, therefore slight adaptations of the presented solution will be necessary depending on the particular instrument.

In the dynamic assessment, we demonstrated that mechanical feedback from the piano action appears in the sensor signal. Finally, the finger sequence measurement serves as an example to show the sensitivity of the sensor to subtle variations in timing and finger pressing force.

We are convinced that the presented sensor is suited for the application in feedback analysis, efficiency and technique studies, as well as for the analysis of playing styles. Therefore, we plan to equip multiple keys of a piano with such sensors. Such a finger pressing force sensing piano keyboard will provide new musical expression possibilities, besides the access to the in-depth investigation of the pianist-piano interaction.

Acknowledgments

This research is pursued as part of the Audio-Haptic Modalities in Musical Interfaces (AHMI) project, funded by the Swiss National Science Foundation (SNSF).

5. REFERENCES

[1] J. O. Smith and S. A. Van Duyne, "Commuted piano synthesis," in *Proceedings of the 1995 International Computer Music Conference, Banff*, 1995, pp. 319–326.

[2] W. Goebl, R. Bresin, and I. Fujinaga, "Perception of touch quality in piano tones," *The Journal of the Acoustical Society of America*, vol. 136, no. 5, pp. 2839–2850, 2014.

[3] A. E. Tiedemann J, Drescher D, "Aus-druck beim klavierspiel: Eine untersuchung zur tastendruckdynamik," *Musikphysiologie und Musiker-Medizin*, pp. 13–21, 2000.

[4] R. A. Moog and T. L. Rhea, "Evolution of the keyboard interface: The bösendorfer 290 se recording piano and the moog multiply-touch-sensitive keyboards," *Computer Music Journal*, vol. 14, no. 2, pp. 52–60, 1990.

[5] D. Parlitz, T. Peschel, and E. Altenmüller, "Assessment of dynamic finger forces in pianists: effects of training and expertise," *Journal of biomechanics*, vol. 31, no. 11, pp. 1063–1067, 1998.

[6] T. Grosshauser and G. Tröster, "Finger position and pressure sensing techniques for string and keyboard instruments." in *NIME*, vol. 13, 2013, pp. 27–30.

[7] A. McPherson, "Portable measurement and mapping of continuous piano gesture." in *NIME*, 2013, pp. 152–157.

[8] ——, "Touchkeys: Capacitive multi-touch sensing on a physical keyboard." in *NIME*, 2012.

[9] H. Kinoshita, S. Furuya, T. Aoki, and E. Altenmüller, "Loudness control in pianists as exemplified in keystroke force measurements on different touches," *The Journal of the Acoustical Society of America*, vol. 121, no. 5, pp. 2959–2969, 2007.

[10] A. Askenfelt and E. V. Jansson, "From touch to string vibrations. i: Timing in the grand piano action," *The Journal of the Acoustical Society of America*, vol. 88, no. 1, pp. 52–63, 1990.

[11] ——, "From touch to string vibrations. ii: The motion of the key and hammer," *The Journal of the Acoustical Society of America*, vol. 90, no. 5, pp. 2383–2393, 1991.

[12] ——, "On vibration sensation and finger touch in stringed instrument playing," *Music Perception: An Interdisciplinary Journal*, vol. 9, no. 3, pp. 311–349, 1992.

[13] H. King, R. Donlin, and B. Hannaford, "Perceptual thresholds for single vs. multi-finger haptic interaction," in *Haptics Symposium, 2010 IEEE*. IEEE, 2010, pp. 95–99.

[14] X.-D. Pang, H. Z. Tan, and N. I. Durlach, "Manual discrimination of force using active finger motion," *Perception & psychophysics*, vol. 49, no. 6, pp. 531–540, 1991.

[15] K. C. Engel, M. Flanders, and J. F. Soechting, "Anticipatory and sequential motor control in piano playing," *Experimental brain research*, vol. 113, no. 2, pp. 189–199, 1997.

[16] T. Grosshauser, G. Tröster, M. Bertsch, and A. Thul, "Sensor and software technologies for lip pressure measurements in trumpet and cornet playing - from lab to classroom," in *Proceedings of SMC*, 2015.

[17] T. Grosshauser and G. Tröster, "Musical instrument interaction: Development of a sensor fingerboard for string instruments," in *Tangible Embedded Interaction, TEI ACM 14*, Munich, Feb. 2014.

[18] W. Goebl and C. Palmer, "Tactile feedback and timing accuracy in piano performance," *Experimental Brain Research*, vol. 186, no. 3, pp. 471–479, 2008.

[19] D. C. Harding, K. D. Brandt, and B. M. Hillberry, "Minimization of finger joint forces and tendon tensions in pianists," *Med Probl Perform Art*, vol. 4, no. 3, pp. 103–108, 1989.

AN EXPLORATION ON WHOLE-BODY AND FOOT-BASED VIBROTACTILE SENSITIVITY TO MELODIC CONSONANCE

Federico Fontana
University of Udine - Italy
Dept. Mathematics,
Computer Science and Physics
federico.fontana@uniud.it

Ivan Camponogara
New York University - Abu Dhabi
Dept. Psychology
ivan.camponogara@gmail.com

Paola Cesari, Matteo Vallicella, Marco Ruzzenente
University of Verona - Italy
Dept. Neurosciences, Biomedicine and Movement Sciences
paola.cesari@univr.it

ABSTRACT

Consonance is a distinctive attribute of musical sounds, for which a psychophysical explanation has been found leading to the critical band perceptual model. Recently this model has been hypothesized to play a role also during tactile perception. In this paper the sensitivity to vibrotactile consonance was subjectively tested in musicians and non-musicians. Before the test, both such groups listened to twelve melodic intervals played with a bass guitar. After being acoustically isolated, participants were exposed to the same intervals in the form of either a whole-body or foot-based vibrotactile stimulus. On each trial they had to identify whether an interval was ascending, descending or unison. Musicians were additionally asked to label every interval using standard musical nomenclature. The intervals identification as well as their labeling was above chance, but became progressively more uncertain for decreasing consonance and when the stimuli were presented underfoot. Musicians' labeling of the stimuli was incorrect when dissonant vibrotactile intervals were presented underfoot. Compared to existing literature on auditory, tactile and multisensory perception, our results reinforce the idea that vibrotactile musical consonance plays a perceptual role in both musicians and non-musicians. Might this role be the result of a process occurring at central and/or peripheral level, involving or not activation of the auditory cortex, concurrent reception from selective somatosensory channels, correlation with residual auditory information reaching the basilar membrane through bone conduction, is a question our preliminary exploration leaves open to further research work.

1. INTRODUCTION

Compared to the perception of auditory pitch, tactile frequency has a less immediate and objective interpretation. Evidence of vibrotactile pitch sensitivity was found decades ago by von Békésy, and ratio codes for pitch have been progressively refined accounting for the complex dependency on the human receptive channels, making pitch a function also of stimulus amplitude, duration, and adaptation of the receptors [1,2,3], furthermore with mechanisms operating at central level [4].

Tactile counterparts of note, consonance, and timbre have been consequently searched, also in deaf people [5], mainly using pairs of sequential stimuli. In a working memory task, Harris *et al.* [6] asked participants to compare the "frequency" of two subsequent vibrotactile square waves to investigate on the retention interval between such stimuli. Evidence of complex tactile waveform discrimination was found by presenting bi-tonal vibrations one after the other while varying the phase of the higher frequency tone [2]. With a similar methodology, varying intensity and/or frequency in the sequential pairs forming the stimulus, a psychophysical model of the Pacinian system inclusive of a critical-band hypothesis of Pacinian coding was presented [3].

In contrast to the use of sequential stimuli, in a recent research work Yoo et al. [7] actuated a graspable haptic device with two sinusoidal vibrations oscillating respectively at a base and chordal frequency, i.e. a carrier and a modulated tone using acoustics terminology, then asking participants to rate the degree of consonance of the resulting vibrotactile stimulus. Their conclusions, obtained after selecting four base and several chordal frequencies set at specific ratios above the respective base, were in favor of a strong dependence of the perceived consonance degree on the beat frequency, which is equal to the absolute difference between chordal and base. The beats, they report, may in fact cause the activation of either the Rapidly Adapting (RA) channel for lower beat frequency values or the Pacinian (PC) channel for higher beat frequency values, respectively responsible for rough as opposed to smoother perceptions and, hence, for an increasing sense of consonance with the chordal vs. base frequency ratio. Similarly to what happens in audition, hence, these results suggest the existence of a link between perceived consonance degree and absolute pitch of the vibrotactile beats, with possible additional sense of tactile roughness this time not caused by cochlear interference, but due to the involvement of separate receptive channels varying with this pitch.

Musical instrument notes define complex stimuli gathering several pure tones with time-varying amplitude together into a harmonic series. In these cases the psychophysical approach is no longer sufficient to completely explain the consonance of a note pair, or interval. Inter-

vals can be ascending or descending if the second note has respectively higher or lower pitch than the first; otherwise they are unisons. Furthermore they are harmonic if the two notes are played simultaneously; melodic if the notes are played sequentially. Experiments involving the vibrotactile presentation of notes investigated the importance of vibrotactile stimuli as an aid to auditory perception [8], as well as showed that Italian and Indian musicians, by touching a harmonium played to reproduce Western and Oriental music scales, were able to disambiguate either scale significantly from the corresponding cutaneous feedback [9]. Concerning musical timbre, participants with no specific musical training, including individuals with auditory impairments, correctly identified vibrotactile presentations of sounds from cello, piano and trombone playing the same note at equal loudness; accuracy in the identification was not lost after substituting these sounds with synthetic ensembles of partial components maintaining the fundamental frequency, temporal envelope and energy of the original sounds, meanwhile changing the spectral centroid [10]. Based on these results the authors suggest an expansion of the tactile critical band model to include musical stimuli, and hypothesize the integration at cortical level of independent cutaneous signals that, besides differentiating among tactile critical bands, furthermore provide somatosensory perceptions resembling acoustic timbre.

While based our methodology on a study by Killam *et al.* [11], who performed a musical interval recognition experiment in which music students were asked to label intervals, both harmonic and melodic, with their standard names in Western music. The authors did not find evidence of a greater difficulty for participants at identifying descending instead of ascending intervals. Furthermore it was observed that, apart from octaves whose simultaneous perception led to precise identifications, melodic intervals were more accurately identified than harmonic. To keep the musical vibrations simple enough as well as their energy centered in the tactile band, we selected the bass guitar for the regularity and compactness of the harmonic series it generates and for the prominence of the component oscillating at the fundamental frequency. Aware of potential bone conduction effects [12], nevertheless we chose to expose participants to vibrations similar to those experienced while attending a musical live performance: stimuli were presented using a vibrating chair along with a vibrating floor platform. Both such actuated objects are not new, used for whole-body vibrotactile stimulation of hearing or deaf individuals [13] and in experiments on foot-based tactile perception [14].

To summarize, participants were first classified as musicians or non-musicians. Both listened to the set of unison, consonant and dissonant intervals prior to the test. Then, every participant had to identify at each trial if an interval was ascending, descending or unison either by hearing, or by feeling vibrations which could be delivered at the whole body or underfoot. Once having checked that the ascending/descending order was not significant (whereas unisons were easier to identify), we categorized the answers based on the consonance degree of the intervals. In addition to the identification task, musicians were asked to label the intervals using their standard name. In the limits of control of this experiment, the results suggest that the subjective ability of both musicians and non-musicians to identify an interval increases with consonance, showing a peak for unisons, and increases if the whole body is stimulated. The musicians' ability to label the intervals varied in a similar fashion.

Yet unpublished so far, these results have been already presented at the HMP Workshop held at the Zürcher Hochschule der Künste in Switzerland[1], where they probably stimulated further research now in progress, appearing somewhere in the future[2].

2. MATERIAL AND METHODS

2.1 Participants

Ten musicians (seven male, three female, age 28.12±6.97; years of practice 12.31±3.43) and eleven non-musicians (nine male, two female, age: 34.33±17.68), all reporting normal hearing and somatosensory ability, participated in the experiment. The study protocol was approved in accordance with the Declaration of Helsinki and all participants gave their written, informed consent now logged at the Department of Neurological and Movement Sciences, University of Verona. Participants were classified as either musicians or non-musicians based on the score they totalized in a questionnaire which is standard in the Italian schools of music. The questionnaire requires knowledge of basic music theory (scales, chords and intervals), the identification of four intervals by listening and finally the ability to sing/play notes and intervals. The group of musicians comprised four classic guitar, three violin, and three cello players.

2.2 Apparatus

A chair and a floor platform were built using thick plywood, and then actuated by mounting a Tactile Sound T239-silver audio-tactile transducer by Clark Synthesis in their bottom part (Figure 1). Both transducers were driven by a Crest CA6 stereo power amplifier duplicating a monophonic sound received from a Macbook Pro laptop driving a Presonus Firebox audio interface. A pair of Sennheiser CX175 in-ear headphones was furthermore connected to the same laptop.

The frequency responses of both the chair and the floor platform were measured under conditions reproducing the experimental test. A 0-dB reference was set in the audio interface corresponding to a loud output from the transducers, just below their distortion threshold. Measurements were made by attaching an Analog Devices model ADXL001 accelerometer to the center of the seat and then to the center of the floor platform: three sessions were made on the chair with sound levels respectively set to -3 dB, -6 dB and -15 dB; then, three sessions were made on the platform with sound levels respectively set to -3 dB, -6 dB and -10 dB. The signals from the accelerometer went through an analog low-pass filter, and

[1] https://www.zhdk.ch/index.php?id=icst_ahmi0
[2] http://www.eurohaptics2016.org/?page_id=1139

were finally acquired by an Arduino 2650 board sampling the signal at 2 kHz. During each session a person weighting 73 kg was sitting on the chair and then standing upright on the floor platform. He was asked to stand still each time the chair and then the platform were excited with a 4 seconds logarithmic sweep in the range 10-1000 Hz through the respective tactile transducer [15]. Three measurements were averaged before closing each session, to attenuate the effects of involuntary body movement on the results.

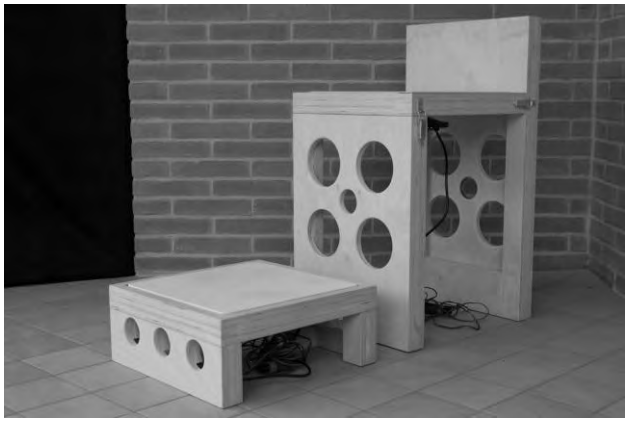


Figure 1. View of the actuated chair and floor platform.

The two averaged frequency responses showed evident spectral similarities, as both plates were made of identical wooden plates mounting the same transducer model; furthermore they were decoupled from the rest of the apparatus, by attaching rubber bands to every corner of the respective plate. Overall these responses showed a linear, though not constant transfer of energy to the body in the frequencies of interest for the experiment.

2.3 Stimuli

Twenty notes were recorded from a Fender Precision electric bass played by a professional musician, and when needed later equalized in amplitude so to have the same loudness at the subjective judgment of the bass player (Table 1). Each note was two seconds long. Such notes were combined into pairs separated by one second of silence, each pair finally representing either an ascending or a unison interval. Twelve intervals were created using these notes. There was no interval larger that the octave.

Note	*Key*	*Frequency (Hz)*	*String*
F#	2	46.25	E
G#	4	51.91	E
A	0	55.00	A
A#	1	58.27	A
B	2	61.74	A
C	3	65.41	A
C#	9	69.30	E
D	5	73.42	A
D#	6	77.78	A
E	7	82.41	A
E	12	82.41	E
F	8	87.31	A
F	13	87.31	E
F	3	87.31	D
F#	14	92.50	E
G	10	98.00	A
G#	11	103.83	A
G#	6	103.83	D
D	17	146.83	A
D#	13	155.56	D

Table 1. Electric bass notes used in the experiment.

The intervals were categorized depending on their consonance level in accordance with the standard Western music notation (Table 2): i) dissonant (4th Augmented, 5th Augmented, 7th Major and 7th Minor); ii) consonant (2nd Major, 4th Perfect, 5th Perfect, and 8th Perfect); iii) unison (the same note repeated twice). Each dissonant and consonant interval was set to be also descending. With the addition of the descending order, eight dissonant (four ascending and four descending), eight consonant (four ascending and four descending) and four unison intervals finally formed the set of stimuli for the experiment.

Category	*Interval name*	*First Note/Key*	*Second Note/Key*
Dissonant	4th Augmented	B/2	F/8
Dissonant	7th Minor	F#/2	E/7
Dissonant	5th Augmented	G/10	D#/13
Dissonant	7th Major	A/0	G#/6
Consonant	4th Perfect	A#/1	D#/6
Consonant	2nd Major	E/12	F#/14
Consonant	8th Perfect	D/5	D/17
Consonant	5th Perfect	C#/9	G#/11
Unison	1st Perfect	G#/4	G#/4
Unison	1st Perfect	C/3	C/3
Unison	1st Perfect	F/13	F/13
Unison	1st Perfect	F/3	F/3

Table 2. Musical intervals used in the experiment.

It must be noticed that identical bass notes, and hence the intervals containing them, could have different spectral content depending on the key pressed to play the note. Table 1 shows that this was the case for the notes E, F, and G#. Composing these notes into melodic intervals led in particular to two unisons, both made with F, which vibrated differently (see Table 2). In particular, the note F had a prominent peak at twice the fundamental frequency when played at the third key. This difference is heard as a

subtle change in musical timbre, of negligible importance for the auditory identification of the interval. As we will discuss later, the same change is significantly discriminated by the tactile system.

2.4 Procedure

Three conditions were defined: 1) in the Feet condition vibrations were delivered by the floor platform to participants standing upright, whereas 2) in the Hips condition participants received vibrations simultaneously under the seat and underfoot while being seated on the instrumented chair. In order to minimize transmission of sounds to the ears, in both such conditions they had to listen to white noise through the earphones meanwhile wearing insulating ear-muffs. Finally, 3) in the Ears condition participants listened to the stimuli directly through the earphones with the vibrotactile transducers switched off; in practice, Ears played the role of a control condition.

Before the test each participant listened to the stimuli, to make her or himself confident with the musical intervals they represented. Then, (s)he was asked to confirm that under conditions Feet and Hips (s)he clearly felt the vibrations resulting from the reproduction of one standard note randomly chosen from the set in Table 1. Finally, each participant was asked to stand about 20 cm far from the chair while wearing the ear-muffs and, then, to set the noise at the earphones to a level preventing him or herself from hearing the sound coming out from the transducers, actually a by-product of practically any (including our) vibrotactile system. With the transducers playing all stimuli in a predefined sequence, every participant was repeatedly asked to increase the loudness until the sound coming from the chair and the floor platform became no longer audible to her or him. This procedure was repeated ten times, and the resulting subjective average individually used for the rest of the experiment.

The test consisted of three experimental blocks respectively implementing the Feet, Hips and Ears condition. Every block was made of a randomly balanced sequence containing six repetitions of each stimulus, for a total of 120 trials. All stimuli were played by means of E-prime by Psychology Software Tools. The order of the experimental blocks was randomized within participants. Resting was allowed amidst each block.

Musicians after each trial had to 1) label the interval with its name, and 2) identify one order for the interval among three possibilities: ascending, descending, unison. Such two questions were ordered by decreasing difficulty to put musicians in condition to answer the more difficult question first. Non-musicians had to answer only question no. 2. It took about 45 to 60 minutes for non-musicians to complete the test. Musicians took more time and comparable fatigue; on the other hand they were more motivated to label each interval and its order. It consequently took 60 to 80 minutes for this group to complete the test.

2.5 Analysis

The percentages of correct answers to question no. 2 under different categories and conditions were considered in the analysis. A preliminary series of t-tests showed that in all such cases these percentages were above chance ($p<0.001$, chance level 100/3 = 33.33%). An ANOVA with repeated measures, having the two groups (musicians and non-musicians) as between factor and the three categories (dissonant, consonant, unison) and conditions (Feet, Hips, Ears) as within factors was performed; pairwise comparisons with Bonferroni corrections were used to explore significant interactions.

Concerning the labeling of intervals, the correct answers to question no. 1 provided by musicians were above chance in percentage ($p<0.05$, chance level 100/12 = 8.33%) except for the dissonant intervals in the Feet condition ($p=0.25$, same chance level). An ANOVA with repeated measures was then conducted on the correct interval labels given by musicians considering the three categories (dissonant, consonant, unison) and conditions (Feet, Hips, Ears) as within factors. Again, pairwise comparisons with Bonferroni corrections were used to explore significant interactions.

3. RESULTS

The analysis of the answers to question no. 2 showed no differences between groups ($F(1,19)=0.13$, $p=0.72$), conversely it showed main differences for category ($F(2,38)=84.36$, $p=0.001$) and condition ($F(2,38)=24.00$, $p<0.001$). Concerning the categories, identifying unisons was significantly easier ($p<0.05$) than identifying consonant and dissonant intervals; as to the conditions, recognition at Feet was significantly less accurate ($p<0.05$) compared to both Hips and Ears (Table 3, rows about identification).

As to question no. 1, an ANOVA with repeated measures conducted limitedly to the group of musicians again showed significant main factors for the category ($F(2,20)=170.50$, $p<0.001$) and condition ($F(2,20)=12.89$, $p<0.001$), as well as a significant interaction for category x condition ($F(4,40)=4.01$, $p=0.008$). Concerning the categories, dissonant intervals were the most difficult to label followed by consonant intervals, while the easiest identification took place with unisons; furthermore, all categories were significantly different each from the other. About the condition, the intervals were identified the least when displaying the stimuli at the Feet, followed by Hips and finally Ears (Table 3, rows about labeling).

	Unison	*Consonant*	*Dissonant*
Identification	95.97±0.77%	93.02±1.06%	88.47±1.91%
Labeling	97.2±8.68%	59.6±4.40%	18.4±6.56%
	Ears	*Hips*	*Feet*
Identification	96.74±0,63%	94.53±0,84%	86.18±1,85%
Labeling	68.1±5.24%	59.3±5.77%	47.8±5.61%

Table 3. Identification and labeling under different categories (above, correct answers) and conditions (below, correct answers).

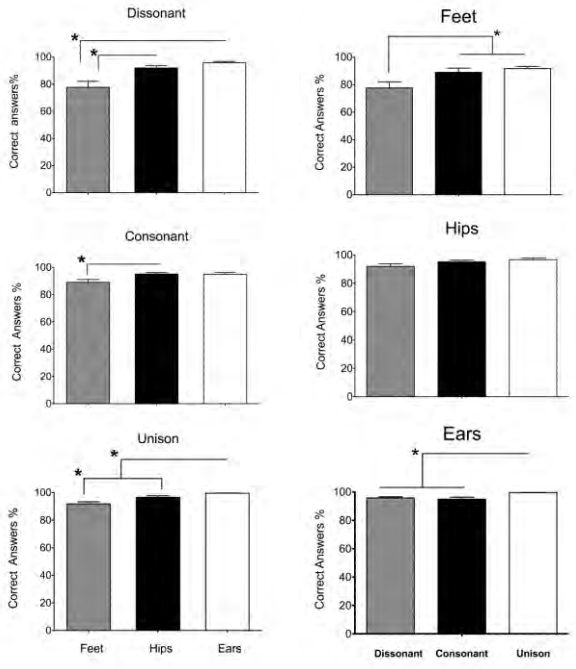


Figure 2. Interval identification: condition x category (left), category x condition (right), all participants. Arrows denote significant differences ($p<0.05$).

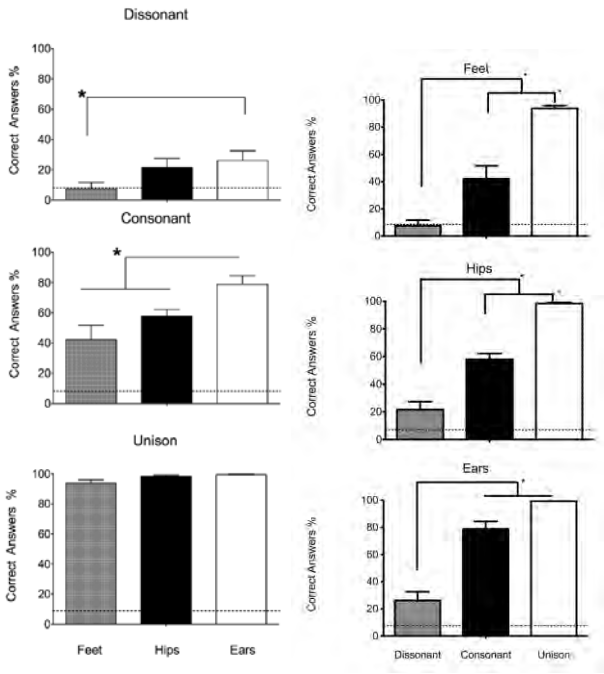


Figure 3. Interval labeling: condition x category (left), category x condition (right), musicians only. Arrows denote significant differences ($p<0.05$).

The interactions between condition and category show significant differences in the identification of the intervals (Figure 2). Differences across categories showed a more difficult identification of consonant as well as dissonant intervals by hearing compared to unisons; instead, their vibrotactile perception led only to significantly more difficult identification of dissonant intervals at foot level. Differences across conditions showed that dissonant and consonant intervals were more difficult to recognize when the stimulus was delivered at Feet compared to when the stimulus was delivered at Hips or Ears; for unisons the three conditions were all different with the highest percentage of correct answers for Ears, followed by Hips and, finally, Feet.

Concerning interval labeling, the condition x category interaction (Figure 3, left) showed that while being exposed to unisons musicians labeled the intervals successfully, with no significant differences depending on the condition; consonant as well as dissonant intervals were instead identified with an accuracy varying with the condition, with significant differences plotted in the figure. In parallel, the category x condition interaction (Figure 3, right) shows that the accuracy was proportional to the categories as well, by progressively decreasing for consonant and then dissonant intervals. In particular, under the Feet condition dissonant intervals were not identified above chance level, in this case equal to 8.3%.

Overall, the interactions in Figures 2 and 3 show that the recognition accuracy decreased along both the category and condition dimensions. The decay was more pronounced when musicians had to label the intervals relying on tactile cues.

3.1 Contribution of different channels

The perception of cutaneous vibrations is known to depend on both the RA and PC channel. The fundamental frequency pairs which defined the intervals forming our stimuli had values falling in the pitch range [46.25 – 155.56] Hz (see Table 1). In this range both channels are stimulated, with a progressively more important contribution of PC for increasing frequency. Since this channel is known to have low sensitivity to pitch, and assuming no adaptation effects in RA under our experimental conditions [2], we might expect a decreasing sensitivity to melodic consonance in our participants for increasing pitch of the intervals.

Based on this expectation, for the unison category we performed an ANOVA considering the two groups as between factor, with the four intervals (see Table 2) and the three experimental conditions as within factors. For the consonant and dissonant categories we classified the interval pitch (see Table 2) as low or high based on the corresponding note fundamental frequency values: consonant 4th Perfect and 2nd Major as well as dissonant 4th Augmented and 7th Minor were classified as low-pitched; consonant 8th Perfect and 5th Perfect as well as dissonant 5th Augmented and 7th Major were classified as high-pitched. Holding this classification, we performed an ANOVA considering, again, the two groups as between factor with pitch (low, high) and conditions (Feet, Hips, Ears) as within factors.

The analysis on unisons showed no differences between groups ($p=0.67$) and a main effect of intervals ($F(3,57)=6.19$, $p<0.0001$) and conditions ($F(2,38)=15.22$, $p<0.0001$).

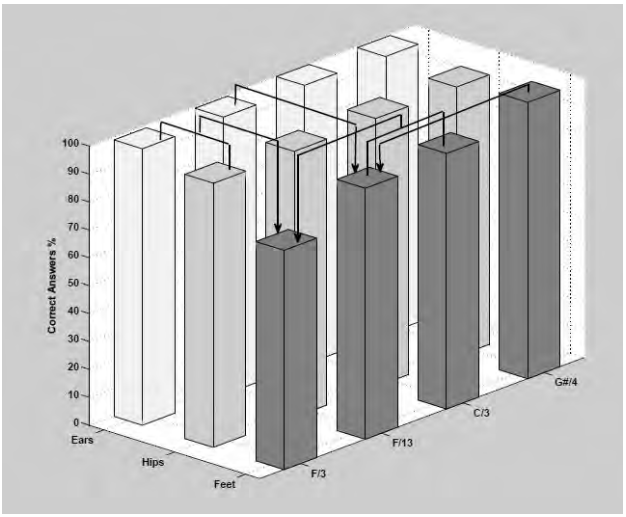


Figure 4. Unison intervals identification: condition x interval, all participants. Arrows denote significant differences ($p<0.05$).

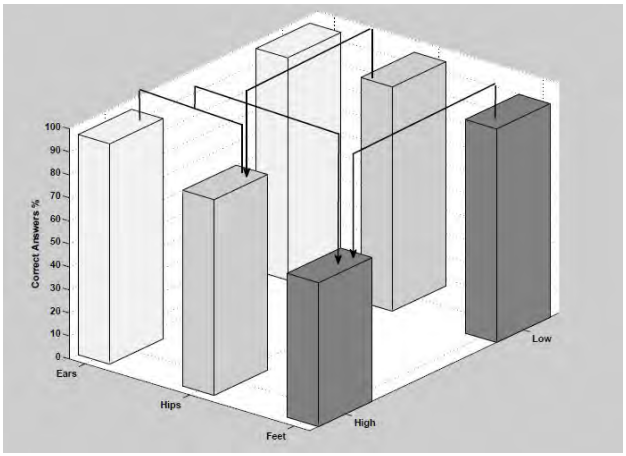


Figure 5. Dissonant intervals identification: condition x pitch, all participants. Arrows denote significant differences ($p<0.05$).

No further main effects or interactions were found. The correct answers were respectively equal to 98.3±0.9% for G#/4, 95.9±1.4% for C/3, 94.9±1.37% for F/13 and 90.7±1.9% for F/3, with significant differences between G#/4 and F/3 as well as C/3 and F/3. Furthermore it showed significant differences in interval x condition ($F(6,114)=6.07$, $p<0.0001$), shown in Fig. 4. This result speaks in favor of a progressively more intense involvement of the PC channel with increasing frequency, with consequent loss of temporal discrimination as suggested by the literature. The same result suggests that the pitch of F/3, although being the same as that of F/13 in auditory sense for what we said in the section on stimuli, was conversely perceived higher in tactile sense, with consequent loss of precision in categorizing the corresponding unison. Finally, the perception was best at the ear and worst underfoot, with significant differences among all the three experimental conditions.

The analysis on the consonant intervals showed no differences between groups ($p=0.66$) and a main effect of conditions ($F(2,38)=9.67$, $p=0.005$), revealing that when the stimuli were delivered underfoot (correct answers equal to 85.8±3.13%) the performance was worse than when they were delivered at the hips (92.6±2.2%) and ears (95.1±1.4%). No further effects or interactions were found. The analysis on the dissonant intervals showed no differences between groups ($p=0.91$) and a main effect of pitch ($F(1,19)=16.481$, $p=0.001$) and conditions ($F(2,38)=38.35$, $p<0.0001$). No further main effects or interactions were found. Participants guessed low-pitched intervals (95.7±1.4%) more correctly than high-pitched intervals (81.1±3.7%). As before, the best perceptual condition was Ears (correct answers equal to 96.2±1.1%) followed by Hips (91.2±2.8%) and Feet (77.7±4.6%). Furthermore it showed significant differences in pitch x condition ($F(2,38)=9.23$, $p=0.001$), shown in Fig. 5.

Taken together, these results indicate that lower-pitched notes were perceived more accurately than higher-pitched notes, with an obvious exception for the Ears condition.

4. DISCUSSION

The performance trend under the tactile categories suggests the existence of an identification process that for some reason worked better when the stimuli were presented to the whole body. The narrower stimulation region associated to the Feet condition might be responsible for the corresponding performance decay.

The two interactions we found once the intervals were classified based on their pitch suggest an extension to the discussion on summation. In fact, the participants' accuracy in classifying unisons across the tactile conditions decayed more rapidly when the pitch of the stimulus was high (Fig. 4); a similar dependence on frequency emerged also when they had to classify dissonant intervals (Fig. 5). Conversely, for low-pitched intervals of any category no significant differences were found across conditions. The logical conclusion is that the RA channel did not need to rely on spatial summation when decoding the intervals falling in its sensitivity range; the PC channel instead benefited from stimulation of larger body areas. This conclusion agrees with results found by Gescheider *et al.* [16] about the summation effects existing for these channels.

An alternative explanation of these results is that some form of auditory perception trough bone conduction took place during the experiment. There are several physiological pathways vibrations can follow to reach the basilar membrane, including those that put the whole cochlea into vibration at audible tactile frequencies [12]: although the auditory masking through white noise of audible by-products coming from vibrotactile devices is standard in these kinds of experiments [1,2,3,7,9,10], in the authors' opinion there is still lack of systematic studies analyzing in detail the potential auditory effects of bone conduction during tactile perception tests, especially when powerful transducers are employed to put the body into vibration in part or as a whole. Since our participants received the stimuli from both transducers only while being seated, we can hypothesize that this condition provided more ener-

getic vibrations to their heads. Furthermore, if we assume that under our experimental conditions bone conduction at low frequencies was more efficient, indeed an hypothesis demonstrated by several researchers [17], then in favor of the same conclusion may also play the pitch-dependent interactions summarized by Fig. 4 and Fig. 5.

There is, however, the concrete possibility that no form of auditory perception took place during the experiment. In this sense the most careful experimental design we were able to find in the literature implemented, besides noisy masking and ear insulation from sounds on air, additional vibrotactile masking through the use of bone headphones delivering white noise to the participant's left and right mastoid [10]. This particular work, for which a concise summary has already been given in the introduction, opens an interesting perspective on experiments that, similarly to our one, aimed at investigating the vibrotactile recognition of musical cues and for this reason disconnected the auditory channel limitedly to sounds on air: Russo and colleagues in fact showed that the use of additional vibrotactile masking did not affect their results.

Since timbre perception implies sensitivity to consonance, the critical band model proposed by Russo *et al.* [10] could explain the sensitivity of our participants once the cortical integration they hypothesize were able to work also across time, by retaining the former note belonging to a melodic interval in the working memory until the latter note was received. As mentioned in the introduction, Harris *et al.* [6] found evidence of this retention using sequential pairs of square waves; since a square wave essentially contains musical cues of dissonance due to its characteristic spectrum, the percentage of correctness (i.e. around 80%) they report for the identification of different retention intervals looks suggestive when compared to the data in our histogram in Fig. 2 about consonance and dissonance perception underfoot, displaying similar percentages.

One further look at Table 1 shows that the fundamentals forming our stimuli ranged quite in the same frequency as the sinusoids used by Yoo *et al.* [7]. While identifying whether an interval was ascending, descending, or unison, our participants were certainly able to decode qualitative tactile cues of relative (that is, either positive or negative) frequency difference between the note fundamentals in the interval. The extent to which this decoding process was also quantitative, hence useful for assessing the consonance degree of a melodic instead of harmonic (e.g. Yoo's) interval, is a question that our experiment cannot answer.

The sensitivity to consonance may have also been influenced by the auditory session our participants attended before the tactile identification tasks. If during these tasks they perceived also with the ears through a pre-attentive or subliminal, however unconscious process caused by bone conduction, the consequent auditory cues may have positively cross-correlated with the vibrotactile stimuli. In fact, a number of research studies have uncovered the existence of cross-modal amplifications which are consequence of the integration, at peripheral and/or central level, of auditory and tactile cues [18,19]. Besides interesting the nervous system from its periphery to the central level, these amplification mechanisms seem to result in greater effects as far as the multimodal stimuli contain components each in relation with the other through consonant frequency ratios [20,21].

We already mentioned that musicians did not perform significantly better than non-musicians while categorizing the vibrotactile stimuli. This evidence reinforces the idea that musicians did not rely on previous musical knowledge while deciding for the increasing or decreasing order of the intervals. Now, once this decision was made, they conversely had to access this knowledge in an aim to label the respective intervals: at this point they labeled consonant intervals well above chance; on the contrary, they were in trouble when associating dissonant intervals to their respective names. Besides an overall offset that affects also the Ears condition, arguing in favor of a moderate difficulty for our musicians to recall the names of the dissonant intervals during the test, once again we speculate that the performance increase obtained with consonant intervals may be the effect of an auditory process occurring when vibrations were produced both under the seat and underfoot, perhaps starting in the basilar membrane due to bone conduction and in possible synergy with a cross-modal amplification process occurring at central level, due to the consonance of the intervals.

5. CONCLUSIONS

Our participants identified the tactile melodic relation between two bass guitar notes with an accuracy that depended on the consonance degree; furthermore, their performance improved by stimulating the whole body. Especially the former dependence reflects a tendency humans experience also when they listen: although dissonance is easy to recognize, consonant cues are inherently preferred. Evidence of this preference has been found in newborns also from deaf parents [22] and in animals [23], bringing researchers to search for universal explanations involving, for instance, geometrical symmetries existing in chords [24] or the existence of forms of entrainment between consonant stimuli and oscillatory neural networks involved in auditory processing [25]. While adding no deeper insight on such explanations, our results at least suggest that humans may process vibrotactile consonance in similar ways they do when they listen to auditory consonance. Since these results were obtained by simulating conditions resembling those that occur when a music performance is attended, we lost much of the control that is instead needed to quantify and explain a perceptual phenomenon. For this reason we remained noncommittal about the possible origins of melodic consonance sensitivity, rather highlighting several affinities existing between our results and a context of previous experimental research from which a robust study of the phenomena underpinning tactile consonance perception could start.

6. REFERENCES

[1] Hollins, E.A. & Roy, M. (1998). A ratio code for vibrotactile pitch. Somatosensory & Motor Research, 15(2), 134-145.

[2] Bensmaïa, S.J. & Hollins, M. (2000). Complex tactile waveform discrimination. Journal of the Acoustical Society of America, 108(3), 1236-1245.

[3] Bensmaïa, S.J., Hollins, M., & Yau, J. (2005). Vibrotactile intensity and frequency information in the Pacinian system: A psychophysical model. Perception & Psychophysics, 67(5), 828-841.

[4] Levänen, S., Jousmäki, V., & Hari, R. (1998). Vibration-induced auditory-cortex activation in a congenitally deaf adult. Current Biology, 8(15), 869-872.

[5] Levänen, S. & Hamdorf, D. (2001). Feeling vibrations: enhanced tactile sensitivity in congenitally deaf humans. Neuroscience Letters, 301(1), 75-77.

[6] Harris, J.A., Miniussi, C., Harris, I.M., & Diamond, M.E. (2002). Transient storage of a tactile memory trace in primary somatosensory cortex. The Journal of neuroscience, 22(19), 8720-8725.

[7] Yoo, Y., Hwang, I., & Choi, S. (2014). Consonance of Vibrotactile Chords. IEEE Transactions on Haptics, 7(1), 3-13.

[8] Darrow, A.A. (1992). The Effect of Vibrotactile Stimuli via the SOMATRON on the Identification of Pitch Change by Hearing Impaired Children. Journal of Music Therapy, 29(2), 103-112.

[9] Romagnoli, M., Fontana, F., & Sarkar, R. (2011). Vibrotactile Recognition by Western and Indian Population Groups of Traditional Musical Scales Played with the Harmonium. In Cooper, E. W., Kryssanov, V., Ogawa, H., & Brewster, S. (Ed.), Haptic and Audio Interaction Design. (pp. 91-100). Heidelberg, Germany: Springer.

[10] Russo, F. A., Ammirante, P., & Fels, D.I. (2012). Vibrotactile discrimination of musical timbre. Journal of Experimental Psychology: Human Perception and Performance, 38(4), 822.

[11] Killam, R.N., Lorton, J., Paul, V., & Schubert, E.D. (1975). Interval Recognition: Identification of Harmonic and Melodic Intervals. Journal of Music Theory. 19(2), 212-234.

[12] Dauman, R. (2013). Bone conduction: An explanation for this phenomenon comprising complex mechanisms. European Annals of Otorhinolaryngology, Head and Neck Diseases, 130(4), 209-213.

[13] Nanayakkara, S., Taylor, E., Wyse, L., & Ong, S.H. (2009). An Enhanced Musical Experience for the Deaf: Design and Evaluation of a Music Display and a Haptic Chair. In Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI'09) (pp. 337-346). New York, NY: ACM.

[14] Cesari, P., Camponogara, I., Papetti, S., Rocchesso, D., & Fontana, F. (2014). Might as well jump: sound affects muscle activation in skateboarding. PloS ONE, 9(3), e90156.

[15] Farina A. (2000). Simultaneous Measurement of Impulse Response and Distortion with a Swept-Sine Technique. Journal of the Audio Engineering Society, 48, 350.

[16] Gescheider, G.A., Bolanowski, S.J., & Verrillo, R.T. (2004). Some characteristics of tactile channels. Behavioural Brain Research. 148(1-2), 35-40.

[17] Paddan, G.S. & Griffin, M.J. (1998). A review of the transmission of translational seat vibration to the head. Journal of Sound and Vibration, 1998, 215(4), 863-882.

[18] Schürmann, M., Caetano, G., Hlushchuk, Y., Jousmäki, V., & Hari, R. (2006). Touch activates human auditory cortex. NeuroImage. 30(4), 1325-1331.

[19] Kayser, C., Petkov, C.I., Augath, M., & Logothetis, N.K. (2005). Integration of Touch and Sound in Auditory Cortex. Neuron, 48(2), 373-384.

[20] Daub, M. & Altinsoy, M.E. (2004). Audiotactile simultaneity perception of musical-produced whole-body vibrations. In Proceedings of the Joint Congress CFA/DAGA. Strasbourg, France: INCE.

[21] Wilson, E.C., Braida L.D., & Reed, C.M. (2010). Perceptual interactions in the loudness of combined auditory and vibrotactile stimuli. Journal of the Acoustical Society of America, 127(5), 3038-3043.

[22] Masataka, N. (2006). Preference for consonance over dissonance by hearing newborns of deaf parents and of hearing parents. Developmental science, 9(1), 46-50.

[23] Watanabe, S., Uozumi, M., & Tanaka, N. (2005). Discrimination of consonance and dissonance in Java sparrows. Behavioural processes, 70(2), 203-208.

[24] Tymoczko, D. (2006). The Geometry of Musical Chords. Science, 313(5783), 72-74.

[25] Large, E. (2011). Musical Tonality, Neural Resonance and Hebbian Learning. In Agon, C., Andreatta, M., Assayag, G., Amiot, E., Bresson, J., & Mandereau, J. (Eds.), Mathematics and Computation in Music (pp. 115-125) Heidelberg: Springer.

David Wessel's Slabs: a case study in Preventative Digital Musical Instrument Conservation

Adrian Freed
University of California, Berkeley
adrian@cnmat.berkeley.edu

ABSTRACT

David Wessel's Slabs is being conserved as an important element of CNMAT's collection of electronic music and computer music instruments and controllers. This paper describes the strategies being developed to conserve the instrument and how we are reaching for the goals of both maintaining the symbolic value of the instrument as a prize-winning, highly-regarded example of the "composed instrument" paradigm and "use value" as an example students and scholars can interact with to develop their own composed instruments. Conservation required a sensitive reconfiguration and rehousing of this unique instrument that preserves key original components while rearranging them and protecting them from wear and damage.

1. INTRODUCTION

We are losing the generation of pioneers of digital computer-based musical instruments – notably most recently Max Mathews in 2011 and David Wessel in 2014. Conserving their existing instruments serves this and future generations of musical instrument makers, performers and composers. It will also yield insight into important questions about the ephemerality of certain approaches to contemporary instrument and controller building.

This paper outlines the plan for conservation of David Wessel's award winning "Slabs" instrument [4]. This represents the current snapshot of a plan that by necessity is in flux as new solutions are sought to the inevitable impacts of entropy. Conservation is a process that never ends, one that requires an explicit plan and documentation to guide future custodians of the instrument who in turn will update the plan.

Our plan is informed by a valuable survey of the ethical and practical issues involved in conserving electronic instruments [2] that emerged from the challenges of conserving and restoring an important collection of Ondes Martenot instruments. Practical issues to be considered include the management of trapped moisture and exposures to light, causes of accelerate aging of metals and plastics respectively.

The first author has had direct experience with such issues when called on to repair ailing Ondes Martenot instruments in the San Francisco Bay Area In one case keyboard contacts were corroded by saline, humid air trapped with the instrument when it was returned to its case on a particularly foggy day [1].

Figure 1: David Wessel with his Slabs

2. SIGNIFICANCE OF THE SLABS

While other digital musical instruments have some of the following valuable qualities, the Slabs is a unique example of an instrument that integrates all of them:

- High temporal resolution (6kHz) and repeatability
- Homogeneous, Many-touch Surface Controller
- Continuous, wide dynamic range of force control
- Played regularly over relatively long period from its predecessor instrument the Buchla Thunder in 1990 until October of 2014.
- Calibration, control structure mappings sound and spatialization programmed by David Wessel qualifying it as a "composed instrument"
- Incorporates 8-channels of audio output for flexible diffusion
- Audio rate gesture representation
- Supports conventional hand drumming gestures and supports, new stroking gestures
- Articulated design aesthetic: "no ceiling on virtuosity"

- Invites and affords practice and rehearsal and coevolutionary design
- Reliable throughout all performances
- Survived numerous airport luggage and security inspection cycles
- Award winning and the subject of many keynote presentations and concert invitations
- Typifies an important design pattern: a custom gesture controller attached by high speed link (Ethernet) to a laptop computer running a media programming language (Max/MSP)

3. GOALS OF CONSERVATION

The Slabs will enter CNMAT's collection of input devices, musical controllers and instruments – a collection maintained of working instruments that support our pedagogical mission and research into the design of future instruments. This diverse collection includes game controllers, computer pointing devices, Buchla Thunder, Lightening, a Mathews Radio Drum, guitars with hexaphonic pickups, midi keyboards and "alternative controllers," and numerous research prototypes. The rarity of some of these holdings and cost constraints means we have to take the position of preventative conservation–not precluded playing the instruments entirely but managing storage, access and use of the instruments to prevent accelerated ageing.

These goals reflect an institutionalized enthusiasm at CNMAT for showing composers and performers how they can use experience with existing instruments to inform the design of new ones.

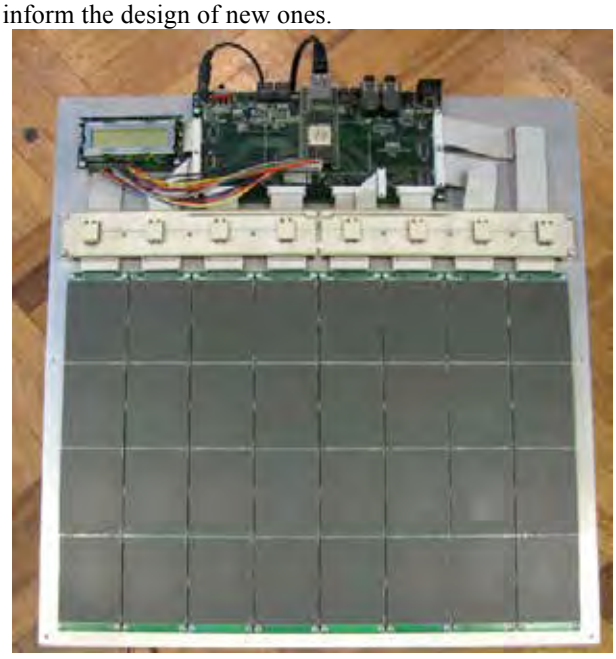


Figure 2. Slabs Controller

4. ANALYSIS

4.1 Macintosh Laptop computer

As a component manufactured in high volume, the original laptop computer from Apple that controls the Slabs is readily conserved by being carefully stored and replaced by a second machine of the same type that has been optimized for longevity in the face of the rigors of performance. The unique component of the original laptop is the bit pattern stored on its hard drive that reflects the bulk of the intellectual effort that went into the instrument including the sounds and control software chosen and written by David Wessel. After being backed up, the data from the original hard drive was transferred to an SSD drive which is inherently more reliable because it has no moving parts and lower power consumption.

To maximize the working life of the new laptop, its hard drive and airport wireless card were removed, it was cleaned of dust and new fans were installed. The SSD drive is powered and installed externally to simplify its replacement and minimize power drawn from the laptop. The following related measures are intended to minimize the accelerated aging that electronic components exhibit at higher temperatures:

- Tuned Energy Saving Settings
- Laptop cooling stand
- Minimizing GPU usage
- Eliminating unnecessary daemons and associated CPU and disk activity unrelated to playing the instrument, e.g., Spotlight.

It is not practical to continuously update the operating system, special Ethernet driver or Max/MSP because of the prohibitive high cost involved and the likelihood the look and feel of the instrument would change. This represents a conservation conundrum because one of the key distinguishing features of digital musical instruments such as the Slabs is that they can be reprogrammed.

Our solution to this conundrum is to provide access to a locked (read only) version of the original Max/MSP software for the machine that supports exploration of the machine in the state David Wessel left it in. Complementing this is a set of baseline patches on which new control structures may be built for the purposes of exploration. Developers of new patches bring their own storage media for this purpose to minimize ageing of the locked SSD drive.

For more ambitious development a patch will be provided that transmits the gestural information from the Slabs as OSC messages to external computers that people can connect up and develop without constraints of legacy operating systems and Max/MSP.

4.2 DAC

Standalone D/A converters with ADAT optical interfaces are readily available as replacements should the original unit fail. After careful dust removal from plug and socket, the optical cable connecting the Slabs to the DAC is plugged in and the seam between plug and socket is sealed with silicone to prevent dust from entering. The silicone can be easily removed and replaced should the cable or DAC need replacement.

4.3 The Slabs Controller

The Slabs controller itself is the most challenging component from the conservation perspective.

4.3.1 The 4x8 Touchpad Array

The VersaPad semiconductive touchpad (by Interlink Electronics) is used to estimate the X and Y position and applied force (Z) of fingers touching its surface. It is usually used as a pointing device in laptop computers and in hand-written signature recognition applications.

The VersaPad consists of a stack of four layers: a base, two sensor layers of resistive film, and a touchpad surface on top. Each sensor layer is made up of two conductive traces (at opposite ends of the device) that are connected to each other through a resistive material. The two sensor layers are rotated 90 degrees with respect to one another, so there is one conductive trace along each edge of the device. These 4 silver, conductive traces are brought to the exterior of the pad by a short length of plastic, flexible flat cable. When an object touches the touchpad the two sensor layers come into contact at the point of applied pressure.

This results in a pressure-dependent resistance between the two layers at the point of contact, and a position-dependent resistance between the point of contact and the conductive traces on each sensor layer. By measuring these resistances it is possible to estimate the location of the point of contact and the amount of force being applied. Interlink usually measures these resistances using an inexpensive PIC based microcontroller circuit (with only a few passive components) to sample the X, Y and Z values 40 times a second. Their patented approach is economical for a single trackpad but did not satisfy the goal for a higher sampling rate of 6kHz and furthermore it does not allow for concurrent sampling of the three measurands. Interlink's approach is to steer currents through the array and infer resistance values from the charge rates of capacitors. This introduces temporal uncertainty and delay which are avoided in a novel circuit by supplying a small constant current through the series-connected X, Y, Z resistors and concurrently converting the induced voltages across these resistors into digital values using a a multi-channel ADC.

4.3.2 Data Acquisition Hardware

The hardware system is made up of nine discrete PC boards shown in Figure 2. Eight of the boards are identical, with 4 touchpads mounted on top and analog conditioning circuitry and multichannel analog-to-digital converters on the bottom. These eight sensor boards are connected to the controller board, which is based around a Xilinx Virtex4 FX12 FPGA. The FPGA has an embedded PowerPC core that runs at 300 MHz. In addition to the FPGA, the controller board has 64M of DRAM, 8M of flash memory, a gigabit Ethernet interface, clock oscillators and a power supply.

4.3.3 Sensor Measurements via OSC

As there are the three X, Y, and Z (force) measurements from each pad there are a total of 32 x 3 = 96 values acquired and transmitted. The design operates in one of two modes. In the first mode, the touchpads are sampled at a low rate (0-200 Hz) and Open Sound Control (OSC) packets containing the measurement data are transmitted as UDP packets over the Ethernet interface. Each OSC packet gives the current X, Y, and Z values only for the pads that are currently being touched.

4.3.4 Sensor Measurements via Audio Signals

The second mode provides for audio sample synchronous output from the pad array. The touchpads are scanned at a one eighth the audio sampling rate (up to 6000 Hz), the data is up-sampled to audio rates (44.1 or 48 kHz), converted to 32-bit floating point, and then encapsulated in a stream of UDP packets which are sent over the Ethernet interface. To avoid performance limitations of the TCP/IP stack on the operating system, a custom driver on the host computer receives these packets and presents them to the operating system as a collection of audio input channels, thereby enabling high-rate control signals in audio processing applications. Max/MSP, provides for up to 512 audio input channels, more than enough for the 96 sensor inputs. This synchronous approach provides for more control intimacy as the gestures are encoded as jitter-free signals locked to the audio sample clock. Sampling the sensors at 1/8 the audio sampling rate results in high temporal resolution (or, equivalently, a wide bandwidth in the frequency domain) on the gestural signals produced by the human performer.

4.3.5 Conserving Baseboards

Only one controller baseboard was built and a key component, the digital FPGA module that sits on it, is no longer manufactured. This module was never made in large quantities and uses parts that have been obsoleted by the manufacturers. Spare parts for the trackpad baseboard and the controller baseboard are available as they were designed at CNMAT.

4.3.6 Conserving the Trackpads

The most troublesome components from the conservation perspective re the Interlink FSR Trackpads. Six months after the construction of the Slabs, it was discovered that the plastic flat cable that carries the four electrical connections from each Versapad is susceptible to brittle plastic failure. The design requires this cable to be bent through 180 degrees to attach to a connector under the pads.

Interlink refused to replace any of the spare parts that were bought ALL of which broke when attempts were made to bend the cable. Interlink claimed that they had never seen this problem due to the special manufacturing techniques they used. We believe the problem re-

sulted from us being suppled old parts from a manufacturing overrun. These had become brittle and that the "special" manufacturing technique that Interlink didn't share with us was simply to bend the cable and insert it into a socket soon after initial manufacturing so that it never needs to be stressed later on as the plastic aged. This theory is supported by the fact that the current pads continue to work and it implies that that they are fragile and their removal would result in cable failure. We plan to build two replacement strips of Versapads using recently manufactured parts that have just entered the distribution channel as spares and eventually it would be wise to build a full set of new strips.

These difficulties with the Versapad turned out to be a productive stimulus for further research that resulted in new designs and new materials for piezoresistive pressure and position sensing surfaces. The recent availability of inkjet-printed conductive and resistive inks suggests that future instruments wouldn't need to be so dependent on a particular manufacturer and its processes. However reproducing the feel of the particular Versapad design used in the Slabs appears to be prohibitively challenging. Professor Wessel learned how to take advantage of the Versapad's strength at capturing low pressure gestures while also avoiding issues of hysteresis, drift and saturation at higher pressures. To learn about such techniques one has to be able to interact with the particular material properties of the Versapads.

Interlink FSR's rely on air gap to avoid the resistive layers sticking to each other from a vacuum that would otherwise form. This suggests that there may be a slow failure mechanism for the pads as dust enters the air gap. Cleaning with bursts of air may be possible but dismantling the pads themselves is not advised because they are so easily damaged in the process. The Versapad surfaces are separated by a thin layer of air and an array of very small, silicone dots which are difficult to repair or replace.

4.4 Stands and Enclosures

The circuit boards constituting the Slabs controller are attached to a solid aluminum base plate that sits (via a bracket) on a foldable floor stand. Although the possibility was often discussed, no case was ever built for the instrument. A small piece of acrylic was bent into a "U" shape and used during shipping to protect the electronics. A sheet of polyethylene foam served to protect the trackpads. Professor Wessel transported the Slabs by wrapping the controller in bubble wrap and packing it into a large polycarbonate suitcase. This is not a tenable arrangement in the long term as we cannot be sure future users of the instrument will be as careful and we can't risk transportation of this unique instrument in checked luggage.

A case is being built that the Slabs can sit in that is designed so that it doesn't interfere with the way the instrument was originally played. A lid will protect the instrument and control dust accumulation while avoiding the possibility of trapped air. An important part of this new case is that it extends sufficiently beyond the back of the instrument to allow for the cables required to configure the instrument to be permanently installed and attached to the case. This relieves wear and tear on cables and connectors and keeps the controller correctly wired to the other components of the instrument. The case includes a shelf below the Slabs to store the DAC, laptop and power supplies. Strong handles are provided so that the entire instrument can be carried from its shelf in CNMAT's musical instrument and controller library to nearby studios for exploratory performance.

Figure 3 The Slabs on collapsible stand

5. CONCLUSION

A wide range of strategies have been presented for conserving the Slabs instrument covering the problems of bit rot, heat accelerated aging, fragilities and wear of mechanical parts and electrical connections and to mitigate the effects of built-in obsolescence. A new case was described that better suits the planned uses of the instruments in the foreseeable future. All these choices were made in the context of the complex ethical issues associated with conservation of musical instruments [3] and in the spirit of Professor Wessel's intention that his instrument be a source of ideas and inspirations for others to design and build their own composed instruments rather than a model instrument to be directly replicated.

Acknowledgments

This conservation project would be impossible without the generous support of Professor Wessel's family, Meyer Sound and anonymous donors supporting research in Professor Wessel's honor.

6. REFERENCES

[1] Madden, D. Advocating Sonic Restoration: Les Ondes Martenot in Practice. *Wi: journal of mobile media 7*(1), 2013.

[2] Ramel, S. Conservation and restoration of electroacoustic musical instruments at the Musée

de la Musique, Paris. *Organised Sound*, *9* (01). 87-90, 2004.

[3] Vidolin, A. Conservation et restauration du patrimoine culturel de la musique électronique. Exemple des Archives Luigi Nono *Journée d'étude: Patrimoine musical du xxe siècle*, Cité de la musique, 2009, 31.

[4] Wessel, D., Avizienis, R., Freed, A. and Wright, M. A Force Sensitive Multi-touch Array Supporting Multiple 2-D Musical Control Structures *New Interfaces for Musical Expression*, New York, 2007.

USING EARSKETCH TO BROADEN PARTICIPATION IN COMPUTING AND MUSIC

Jason Freeman
Georgia Institute of Technology
jason.freeman@gatech.edu

Brian Magerko
Georgia Institute of Technology
magerko@gatech.edu

Doug Edwards
Georgia Institute of Technology
doug.edwards@ceismc.gatech.edu

Morgan Miller
SageFox Consulting
mmiller@sagefoxgroup.com

Roxanne Moore
Georgia Institute of Technology
roxanne.moore@ceismc.gatech.edu

Anna Xambó
Georgia Institute of Technology
anna.xambo@coa.gatech.edu

ABSTRACT

EarSketch is a STEAM learning intervention that combines a programming environment and API for Python and JavaScript, a digital audio workstation, an audio loop library, and a standards-aligned curriculum to teach introductory computer science together with music technology and composition. It seeks to address the imbalance in contemporary society between participation in music-making and music-listening activities and a parallel imbalance between computer usage and computer programming. It also seeks to engage a diverse population of students in an effort to address long-standing issues with underrepresentation — particularly of women — in both computing and music composition. This paper summarizes the design of the EarSketch curriculum and learning environment and its deployment contexts to date, along with key findings from a pilot study. It builds upon prior publications by contextualizing the project's motivations and interpreting its findings in the dual realms of participation in computer science and in music creation.

1. INTRODUCTION

Music has increasingly become a commodity to be heard rather than a creative experience in which to partake. Recent data from the National Endowment for the Arts in the United States (Figure 1) shows that only a small percentage of American adults engage in music-making activities even once per year, while a far greater percentage listen to recorded or live music [1].

In the field of computing, a similar divide is evident between using computers or smartphones and programming those same devices (Figure 2). Change the Equation expresses this divide succinctly in arguing that "digital native does not mean tech savvy: 83% of millennials say they sleep with their smartphones, yet 58% of millennials have poor skills in solving problems with technology" [2]. This relative lack of computational skills is more than an economic problem, with a growing demand for computing jobs in the workforce [3]. Just as music has long been a core mechanism for human expression and collaboration [4], computing is becoming a core 21st century skill: understanding the algorithms behind computers and how to write code is essential to understanding the benefits, grappling with the limitations, and harnessing the creative potential of new computing technologies [5].

Creating Music	
Played a musical instrument, alone or with others	12%
Sang, either alone or with others	9%
Created or performed music	5%
Recorded, edited, remixed musical performances	4%
E-mailed, posted, or shared one's own music	3%
Used a computer, a handheld or mobile device, or the Internet to create music	1%
Consuming Music	
Used TV, radio, or the Internet to access music of any kind	57%
Used a handheld or mobile device to access music of any kind	34%
Attended a live music performance of any kind	32%

Figure 1. Data from the US National Endowment for the Arts on the percent of American adults (18 years and older) who have engaged in various music activities at least once over a 12-month period [1].

In the academy, computing and music have an additional commonality: both fields struggle with gender imbalance. Computer science has well-documented challenges with underrepresentation of women at all stages of the pipeline [6], with the problem generally worsening in recent decades even as other disciplines have improved [7]. Music theory and composition is one of the few academic disciplines in which an even smaller percentage of PhDs are earned by women than in computer science [8].

Coding	
"do programming" at work [9]	15%
K-12 schools offering CS courses with programming in the US [10]	25%
Students who are very likely to learn [more] computer science in the future [10]	27%
Using Computers	
Owns a desktop of laptop computer [11]	73%
Owns a smartphone of some kind [11]	68%
Uses at least one social networking site [12]	65%

Figure 2. Data on computer usage and coding.

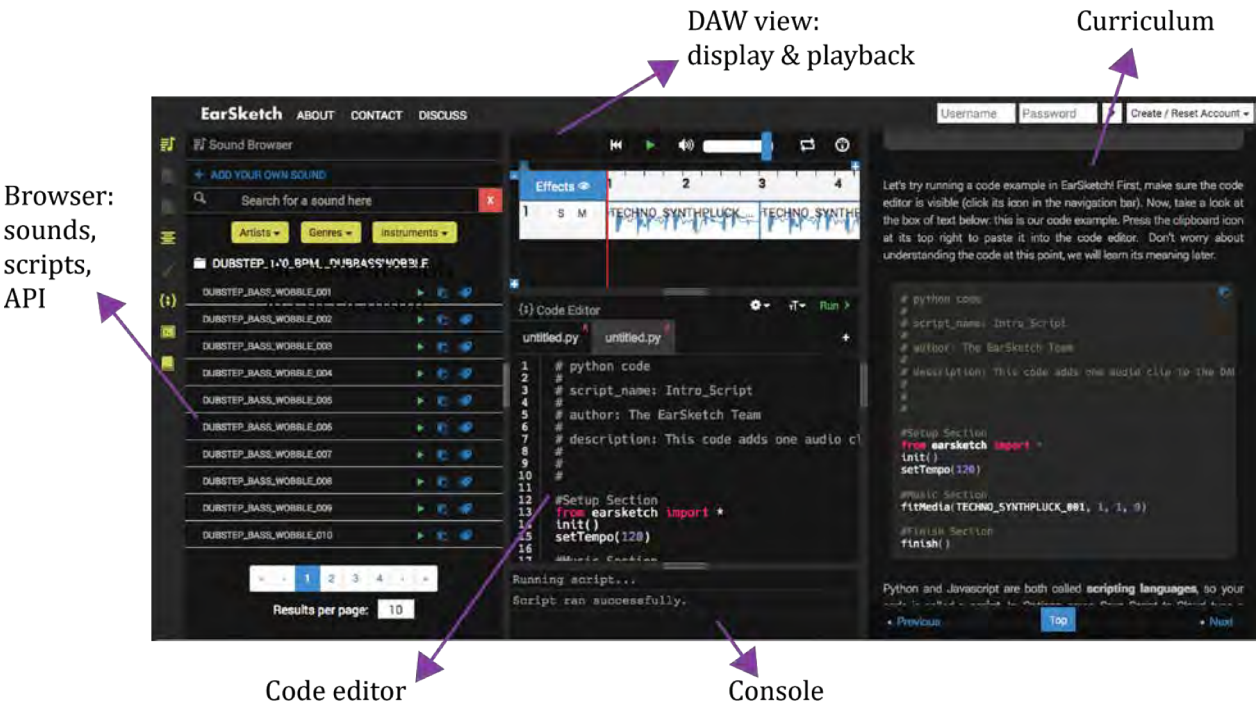


Figure 3. The EarSketch web-based learning environment.

It is in this context that we developed EarSketch (Figure 3), an integrated STEAM [13] programming environment, digital audio workstation, loop library, and curriculum that teaches elements of computing and music together in order to engage a diverse population of teenage students in both domains. EarSketch seeks to address the divides between music consumption and creation and between computer usage and computer programming in a manner that engages populations traditionally underrepresented in these fields.

This paper contextualizes EarSketch in related work, outlines core design principles of the project, describes the learning environment and its components in detail, summarizes the educational contexts in which it has been used, reviews results from pilot studies in schools, and outlines areas of current and future work for the project.

2. RELATED ENVIRONMENTS

EarSketch is inspired by numerous learning environments that have combined computing and music to facilitate learning in both domains through an algorithmic approach. For example, MediaComp [14] teaches introductory Python programming in part by teaching students how to implement simple audio effects. Sonic Pi [15] focuses on live coding on an embedded computing device for both sound synthesis and symbolic music generation. Performamatics [16] brings together computer science and music students to create interactive musical instruments through visual programming paradigms. And JythonMusic [17] focuses on the algorithmic generation of symbolic scores and on real-time communication with other software and devices.

In addition to these and other specialized programming environments and curricula, many popular computer music languages, such as Max [18], ChucK [19], SuperCollider [20], and Faust [21], are used pervasively in music technology pedagogy in university courses. And many programming environments designed specifically for computing education, such as Scratch [22] and PencilCode [23], include functionality for recording and playing back sound and for generating MIDI note data.

EarSketch is distinct from these other environments in three significant ways. First, EarSketch relies neither on a knowledge of symbolic music representation (e.g. MIDI note numbers or note names as in [16], [17], [18], [22], and [23]) nor on a knowledge of audio synthesis or signal processing techniques (e.g. unit generators or sample-level manipulation as in [14], [15], [18], [19], [20], and [21]).

Second, EarSketch exists primarily within the paradigm (and interface of) a digital audio workstation. Unlike Max for Live [24], the integration is not primarily through effects and plugins but rather through programmatic operations on the multi-track DAW timeline itself. In this way, EarSketch is closest in lineage to the ReaScript Python API found within the Reaper DAW [25]. (In fact, early versions of EarSketch were implemented within Reaper using ReaScript.)

Third, EarSketch focuses on enabling users to quickly create complete songs with short scripts and limited (but developing) musical and computational skills. This low barrier of entry, combined with simple design patterns to quickly create hierarchical musical structures, is meant to drive immediate engagement with music and coding for our young, novice, and diverse target audience.

3. CORE DESIGN PRINCIPLES

To further explore these distinct features of EarSketch, we now discuss five guiding design principles: creative and personal expression; real-world connections; accessibility to beginners; music-driven computational learning; and standards-based curriculum.

3.1 Creative and Personal Expression

Historically, introductory computer science has been taught as a series of abstract problems to be solved. Common student assignments ask students to sort words in a list or print out a number sequence [26].

Music, as taught to teenagers in band and orchestra classes, often focuses on rote reproduction of notated rhythms and pitches, with minimal emphasis on student creativity and expression [27]. New technologies such as SmartMusic [28] further emphasize this focus by grading students solely on how well they reproduce the notated elements of music.

EarSketch, in contrast, emphasizes open-ended assignments with no "correct" answers. Students compose music algorithmically by writing code. Their work must abide by a set of broad musical and computational constraints (e.g. to use a loop, or to incorporate at least three tracks), but students exercise wide artistic freedom and write in a wide variety of musical styles. We hope that students create music that meaningfully represents them, that they like, and that they wish to share. This approach is inspired by constructionism [29] and by studio-based learning as found in art and architecture courses [30].

3.2 Real-World Connections

To help drive student motivation, we wanted EarSketch to feel relevant to the computing and music industries.

In computing, EarSketch teaches students popular real-world programming languages. (They choose between Python or JavaScript.) This also enables students to transfer coding skills directly to other domains and contexts.

In music, EarSketch adopts the paradigm of a digital audio workstation (DAW). The user interface mimics the look and feel of popular DAWs with multi-track audio and effects lanes. The application programming interface (API) for JavaScript and Python mirrors this functionality, with core API functions to support placement of audio on the multi-track timeline, step-sequencing, and control over effects parameters and automations. The audio loop library itself provides an additional real-world connection, as the loops were created by music industry veterans: Richard Devine, an experimental electronic musician and commercial sound designer; and Young Guru, an audio engineer best known for his work with Jay-Z.

This focus on real-world connections is inspired by the notion of thick authenticity [31]. The authenticity of a learning experience, according to [32], is based on the interrelated authentic learning practices of: a) having personally meaningful learning experiences; b) learning that relates to the world outside of the learning context; c) learning that encourages thinking within a particular discipline (e.g. music composition); and d) allowing for assessment that reflects the learning process. Thick authenticity, according to [31], meets all of these requirements in a single approach / system.

3.3 Accessibility to Beginners

EarSketch was intended for widespread use amongst student and teacher populations with limited (if any) prior experience in computing or music. We therefore designed the learning environment to require no prerequisite skills in either domain.

In the computational domain, we focused our curriculum on beginning programming concepts, such as variables, functions, loops, conditionals, and lists.

In the musical domain, we teach basics of musical time and form (tempo, rhythm, measures, structures such as ABA and verse-chorus, etc.), avoid references to pitch and chord names and music notation, and structure the audio loop library as a collection of sound packs that are designed to naturally fit well together, and focus more on the hierarchical level of audio loops than on individual musical events.

3.4 Music-Driven Computational Learning

EarSketch's grounding in digital audio workstations invites comparison to commercial music production software. There is a risk that students may be unmotivated to learn new computational concepts or to write code if they can easily achieve similar results in a traditional DAW.

We addressed this challenge by always introducing new computational concepts in service of musical ends, showing how code can sometimes create music more quickly and easily than a graphical interface, how it can enable musicians to rapidly experiment with many different musical alternatives, and how it can enable the use of musical techniques that would be impossible to achieve in a traditional DAW.

One example of this approach is the use of strings to create and vary drum beats. We introduce a string notation for step-sequencing, inspired by ixi.lang [34] and LOLC [35], in which each character represents a sixteenth-note sound hit, tie, or rest. Once these strings are created, they can be modified with string operations to be repeated, concatenated, split, shuffled, and otherwise modified. This introduces students to the notion of music as a balance between repetition and variation while providing them with the specific technique of string creation and manipulation to actualize this concept in music.

3.5 Standards-Based Curriculum

To facilitate widespread adoption of EarSketch in learning contexts aimed at our target age demographic, we focused on high-school computer science classrooms and on a new curriculum standard in the United States: Computer Science Principles [33].

Our curriculum, and by extension core features of the EarSketch API and learning environment, were therefore designed specifically to address the learning objectives in the Computer Science Principles framework. For example, EarSketch focuses primarily on imperative programming paradigms and on constructs for iteration, abstraction, and branching that fit within such paradigms, while avoiding object-oriented structures or functional approaches that, while used widely in music technology (e.g. [20]), are not emphasized in the Computer Science Principles framework.

4. THE LEARNING ENVIRONMENT

EarSketch is a free, web-based environment that integrates multiple components within a single browser window [36]. In this section, we describe its main components: the programming environment, the digital audio workstation (DAW), the loop library, and the curriculum.

4.1 Programming Environment

In the EarSketch code editor, students write code in Python or JavaScript, using either a text editor or a blocks-based visual code editor [37]. Regardless of language or editor chosen, they use the same application programming interface (API) to create music.

Figure 4 shows a simple EarSketch program. `fitMedia()` places an audio clip on a particular track and starting/ending times, looping or truncating the clip as necessary to fill the specified amount of time. `makeBeat()` step-sequences a rhythm, with each character of a string representing a sixteenth-note: a "0" plays a sound file from the beginning, another digit plays alternate sound files at other indices within a list, a "+" ties (or continues playing) the sound file, and a "-" makes a rest (silence). `setEffect()` adds an effect to a track (or the master track), with optional parameters to specify effect parameters and to define an envelope for those parameters.

```
from earsketch import *

init()
setTempo(120)

fitMedia(HOUSE_ROADS_PIANO_007, 1, 1, 3)
setEffect(1, VOLUME, GAIN, -60, 1, 0, 3)

beatElement = OS_LOWTOM01
beatString = "0+++0+++0+0+0+0+"
for index in range(1,3):
   makeBeat(beatElement,2,index,beatString)

finish()
```

Figure 4. A sample EarSketch Python script that places an audio clip on track 1, adds a volume effect automation to track 1, and places a step-sequenced beat on track 2.

Figure 5 shows a more complex example that mimics the practice of hocketing to create a hybrid drum track out of two audio sources. For each sixteenth note in the timeline, the RMS amplitude of each track is computed. The louder track's level is then set to 0 dB for that sixteenth note, and the quieter track's level is set to -60 dB.

Additional API methods offer alternate methodologies to audio file placement and implement utility functions such as console and file input and string manipulation.

By default, EarSketch operates in a batch mode. Code is interpreted when hitting the "run" button to create the music. It does not run interactively while the music is playing. This approach follows a music production methodology which is focused more on creating a fixed-media track than on live performance. EarSketch does support live coding [38]. Users can write and execute code while audio is playing, and audio playback will update seamlessly.

```
from earsketch import *
init()
setTempo(120)

sound1 = ELECTRO_DRUM_MAIN_BEAT_001
sound2 = ELECTRO_DRUM_MAIN_BEAT_002
analysisMethod = RMS_AMPLITUDE
hop = 0.0625 # analyze 1/16th note chunks
start = 1
end = 3.0

fitMedia(sound1, 1, start, end)
fitMedia(sound2, 2, start, end)

position = 1
while (position < end):
   feature1 = analyzeTrackForTime(1,
      analysisMethod, position,
      position + hop)
   feature2 = analyzeTrackForTime(2,
      analysisMethod, position,
      position + hop)
   if (feature1 > feature2):
      setEffect(1, VOLUME, GAIN, 0,
         position, 0, position + hop)
      setEffect(2, VOLUME, GAIN, -60,
         position, -60, position + hop)
   else:
      setEffect(1, VOLUME, GAIN, -60,
         position, -60, position + hop)
      setEffect(2, VOLUME, GAIN, 0,
         position, 0, position + hop)
   position = position + hop

finish()
```

Figure 5. An algorithmic EarSketch example in which two tracks' mute states are toggled every sixteenth note depending on which has the higher amplitude.

4.2 Digital Audio Workstation

The digital audio workstation panel within EarSketch displays the visual output of code execution in a standard multi-track format. It is not a fully-functional DAW in that students cannot add, edit, or delete audio clips or effects; this must be done through coding. Students can navigate their project by using transport controls, soloing/muting tracks, and bypassing effects. They can also export their project as a stereo mixdown (WAV, MP3, or Soundcloud) or a multi-track project to continue editing in a traditional desktop DAW.

4.3 Loop Library

EarSketch includes ~4000 loops accessible via a sound browser sidebar. The sound browser pane mimics the functionality of similar interface panels in DAWs, allowing users to search and filter sounds by artist, genre, and instrument. Sounds are grouped into collections that contain loops designed to fit well together. Users may also upload their own sounds from their computer or quick-record new sounds directly within EarSketch.

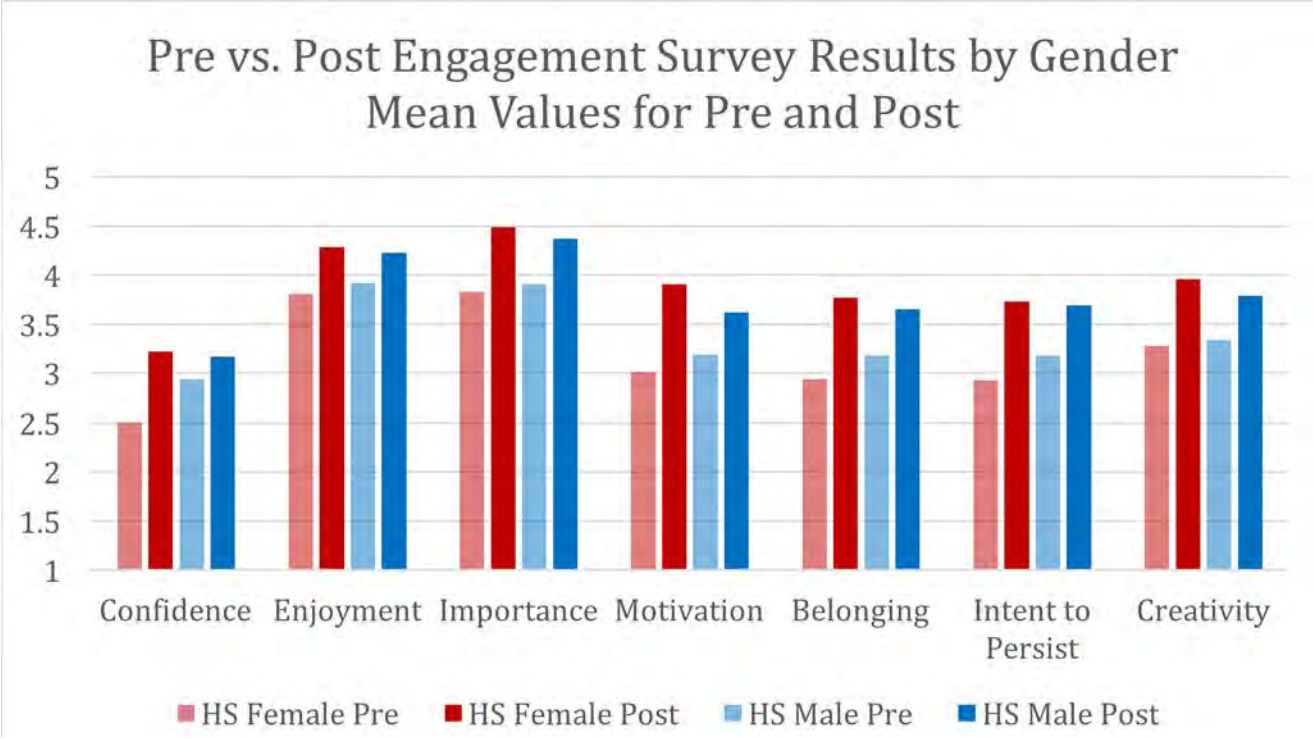


Figure 6. Pre and post engagement survey results across male and female students from a 2013 EarSketch pilot study. Across all seven engagement constructs, female students are less engaged at pre than their male counterparts but more engaged at post than the male students.

Each sound in the library is identified by a unique constant. To use the sounds within EarSketch, users simply paste the constants into the code editor as function arguments. EarSketch automatically time-stretches loops to match the overall project tempo.

4.4 Curriculum

The EarSketch curriculum is intended to be used within introductory computing and music technology courses, and is specifically aligned to AP Computer Science Principles [33], an emerging curriculum standard in the United States for computer science courses at the high school (teenage) level. The EarSketch curriculum covers computational topics such as data types, variables, functions, lists, loops, boolean logic, conditionals, and strings, and music and music technology concepts such as DAW basics, musical form, rhythm, meter, tempo, and texture.

A sidebar within EarSketch displays textbook-like materials for students to use for self-study and as a reference: this includes text, runnable code examples, video demonstrations, and slides. Classroom instructors can access teaching materials that include day-by-day lesson plans, handouts, and projects and assessments.

Each summer since 2014, the EarSketch team has conducted professional learning workshops for teachers interested in adopting EarSketch. These workshops teach EarSketch and the music and computing fundamentals teachers need for the course, as well as pedagogical techniques on topics such as facilitating student collaboration, discussing student projects, and assisting students in debugging code.

5. DEPLOYMENT CONTEXTS

EarSketch has been used in a variety of educational contexts, including academic courses in computing and in music technology at high schools; summer camps for middle school and high school students; undergraduate-level introductory computing courses; and a Massive Open Online Course (MOOC) in music technology taught by one of the authors (Freeman) on Coursera [39].

Since we launched the web based version of EarSketch in 2014, over 55,000 unique users from over 100 countries have coded with EarSketch, saving over 47,000 projects to our server.

6. PILOT RESULTS

Between 2013 and 2015, we have conducted multiple pilot studies of EarSketch in academic computing courses at four different Atlanta-area high schools. To study the impact of EarSketch on students in these courses, we employed a variety of research methods, including questionnaires, content assessments, observations, interviews, and focus groups. The content knowledge assessment, given before and after the EarSketch module of the course, was a multiple-choice assessment aligned to the learning objectives of the courses.

A student engagement survey, administered retrospectively, monitored potential changes in students' internal characteristics. This instrument draws scales from Williams, Weibe, Yang, & Miller [40] and Knezek & Christensen [41] measuring computing confidence, computing enjoyment, computing importance and perceived usefulness, motivation to succeed, and computing identity and

belongingness as predictor variables and an intention to persist in computing as an outcome variable. The literature in STEM education suggests that these constructs are critical to enhancing the number of under-represented students who persist in STEM fields [40]. Earlier versions of the instrument also adopted the Creativity Support Index [42], but more recently we have developed our own questions to gauge students' perceptions about creativity, building on prior research [43], [44] on creativity.

We now summarize findings of our 2013 pilot study [45], in which we compared results of male and female EarSketch students. The study included students in two courses. One was an introductory computing course; the other was an introductory music technology course. Both included a similar EarSketch curricular module. 97 students provided usable data across all student survey constructs, with 27% of them female and 73% male (a typical breakdown for these courses). Students did not know they would be using EarSketch prior to course enrollment.

Both male and female students showed statistically significant increases from pre to post across all engagement constructs ($p < 0.01$), with the exception of male confidence ($p = 0.07$). Furthermore, female students expressed greater pre-to-post change across all constructs than male students; these differences are significant ($p < 0.05$) in confidence, motivation, and identity and belongingness. Figure 6 shows this visually: across all constructs, female students are less engaged than their male counterparts before they study EarSketch but are more engaged than the male students after studying EarSketch. Both male and female students' content knowledge also significantly increased from pre to post, but there were no significant differences between male and female student gains in this area. These results are discussed in more detail in [45].

Focus groups and free-response items in student surveys suggest that the core design principles guiding EarSketch play an important role in student engagement. Some students remarked on the importance of personal expression and creativity, commenting that "I got to express my ideas and it was fun and inspiring to see that I could be good at computing" and "I enjoyed making my own music tracks that people, including myself, actually liked." Others focused on the importance of real-world context, noting that "I liked learning how music is made and how we can learn and get good at doing things that people in the music industry do now." This seemed to impact students' interest in persisting in further study, as evidenced by this comment from a focus group: "It gives me choices for college. Like this is something I would actually like to do for college and I'd actually like to do probably with my life. Yeah. I would love to do it."

Our pilot studies have been more focused on computer science content knowledge and attitudes than on music, but in a fall 2015 pilot study, we did collect data on students' experiences with music prior to the course. 83% of students stated that they listened to music for at least one hour every day. Only 6% of students had mixed or composed their own music prior to using EarSketch, and only 36% were involved in music performance activities such as band, chorus, orchestra, or instrumental lessons. This imbalance between music listening and music making mirrors the data from the US National Endowment for the Arts. In future studies we hope to measure if and how EarSketch has engaged students in music making beyond the course and beyond EarSketch itself.

These pilot results suggest that EarSketch has strong potential to engage students — and particularly female students — in computing and music at the introductory level.

7. DISCUSSION AND FUTURE WORK

Adoption of EarSketch is growing rapidly, and we are currently scaling up our research efforts to understand how EarSketch impacts student engagement and content knowledge across diverse populations and school contexts. Over the next two years, we will be expanding our research efforts to study EarSketch in approximately 30 high school AP Computer Science Principles classrooms in Georgia, using content knowledge assessments, engagement surveys, observations, interviews, and focus groups to understand how EarSketch impacts students and how we can continue to improve it. As part of this study, we are also comparing classrooms using EarSketch to classrooms using other learning environments. We are also using complex systems modeling techniques [46] to model the complex sets of attributes and relationships that underlie learning interventions.

At the same time, we are expanding EarSketch to new modalities and to new learning contexts. We are continuing to develop a blocks-based visual programming editor for EarSketch that will enable us to more successfully incorporate it into classrooms with younger students. We are also currently developing a collaborative, tabletop interface suited for museum installations and outreach events. We recently added support for P5 [47] into EarSketch to support the generation of live visuals alongside music, and are exploring ways to connect EarSketch to physical computing systems such as the Moog Werkstatt [48] and the Lilypad Arduino [49]. We are also interested in finding ways to integrate EarSketch into other computer music and general-purpose programming environments.

Regardless of the modality and context, our goals remain the same: to engage a broad and diverse population in making music and writing code, and in doing so to spark their interest in these activities such that they persist and continue to develop beyond a single learning intervention.

Acknowledgements

The EarSketch project receives funding from the National Science Foundation (CNS #1138469, DRL #1417835, and DUE #1504293), the Scott Hudgens Family Foundation, the Arthur M. Blank Family Foundation, and the Google Inc. Fund of Tides Foundation. EarSketch is available online at http://earsketch.gatech.edu. A com-

plete list of personnel on the EarSketch team is available at http://earsketch.gatech.edu/landing/team2.html.

8. REFERENCES

[1] S. Iyengar, "How a Nation Engages with Art: Highlights from the 2012 Survey of Publication Participation in the Arts," National Endowment for the Arts, 57, 2013.

[2] Change the Equation, "Digital Native Does Not Mean Tech Savvy," 11-Jun-2015. [Online]. Available: http://changetheequation.org/stemtistics/digital-native-does-not-mean-tech-savvy. [Accessed: 25-Mar-2016].

[3] Code.org, "Promote Computer Science," *Code.org*. [Online]. Available: https://code.org/promote. [Accessed: 25-Mar-2016].

[4] I. Peretz, "The nature of music from a biological perspective," *Cognition*, vol. 100, no. 1, pp. 1–32, May 2006.

[5] J. M. Wing, "Computational thinking," *Commun ACM*, vol. 49, no. 3, pp. 33–35, 2006.

[6] J. Margolis, "Unlocking the Clubhouse: A Decade Later and Now What?," in *Proceeding of the 44th ACM Technical Symposium on Computer Science Education*, New York, NY, USA, 2013, pp. 9–10.

[7] National Science Foundation, "Science and Engineering Degrees: 1996-2010," 2013. [Online]. Available: http://www.nsf.gov/statistics/nsf13327/content.cfm?pub_id=4266&id=2. [Accessed: 25-Mar-2016].

[8] S.-J. Leslie, A. Cimpian, M. Meyer, and E. Freeland, "Expectations of brilliance underlie gender distributions across academic disciplines," *Science*, vol. 347, no. 6219, pp. 262–265, Jan. 2015.

[9] C. Scaffidi, M. Shaw, and B. Myers, "Estimating the numbers of end users and end user programmers," in *2005 IEEE Symposium on Visual Languages and Human-Centric Computing*, 2005, pp. 207–214.

[10] Gallup, "Images of Computer Science: Perceptions Among Students, Parents, and Educators in the U.S.," Google, 2015.

[11] M. Anderson, "Technology Device Ownership: 2015," *Pew Research Center: Internet, Science & Tech*, 29-Oct-2015. .

[12] A. Perrin, "Social Media Usage: 2005-2015," *Pew Research Center: Internet, Science & Tech*, 08-Oct-2015. .

[13] J. Maeda, "STEM+ Art = STEAM," *STEAM J.*, vol. 1, no. 1, p. 34, 2013.

[14] M. Guzdial, "A media computation course for non-majors," *ACM SIGCSE Bull.*, vol. 35, pp. 104–108, 2003.

[15] S. Aaron and A. F. Blackwell, "From Sonic Pi to Overtone: Creative Musical Experiences with Domain-specific and Functional Languages," in *Proceedings of the First ACM SIGPLAN Workshop on Functional Art, Music, Modeling & Design*, New York, NY, USA, 2013, pp. 35–46.

[16] J. M. Heines, G. R. Greher, and S. Kuhn, "Music Performamatics: Interdisciplinary Interaction," in *Proceedings of the 40th ACM Technical Symposium on Computer Science Education*, New York, NY, USA, 2009, pp. 478–482.

[17] B. Manaris and A. R. Brown, *Making Music with Computers: Creative Programming in Python*, vol. 13. Boca Raton, FL: CRC Press, 2014.

[18] M. Puckette, "Combining event and signal processing in the MAX graphical programming environment," *Comput. Music J.*, pp. 68–77, 1991.

[19] G. Wang, P. R. Cook, and S. Salazar, "ChucK: A Strongly Timed Computer Music Language," *Comput. Music J.*, vol. 39, no. 4, pp. 10–29, Dec. 2015.

[20] J. McCartney, "Rethinking the Computer Music Language: SuperCollider," *Comput. Music J.*, vol. 26, no. 4, pp. 61–68, Dec. 2002.

[21] Y. Orlarey, D. Fober, and S. Letz, "Syntactical and semantical aspects of Faust," *Soft Comput.*, vol. 8, no. 9, pp. 623–632, Jul. 2004.

[22] M. Resnick, J. Maloney, A. Monroy-Hernández, N. Rusk, E. Eastmond, K. Brennan, A. Millner, E. Rosenbaum, J. Silver, and B. Silverman, "Scratch: programming for all," *Commun. ACM*, vol. 52, no. 11, pp. 60–67, 2009.

[23] D. Bau, D. A. Bau, M. Dawson, and C. Pickens, "Pencil code: block code for a text world," in *Proceedings of the 14th International Conference on Interaction Design and Children*, 2015, pp. 445–448.

[24] V. J. Manzo and W. Kuhn, *Interactive Composition: Strategies Using Ableton Live and Max for Live*. Oxford: Oxford University Press, 2015.

[25] J. Frankel, "ReaScript," 2005. [Online]. Available: http://www.reaper.fm/sdk/reascript/reascript.php. [Accessed: 01-Jan-2015].

[26] L. Layman, L. Williams, and K. Slaten, "Note to self: make assignments meaningful," *ACM SIGCSE Bull.*, vol. 39, no. 1, pp. 459–463, 2007.

[27] R. E. Allsup, "Mutual learning and democratic action in instrumental music education," *J. Res. Music Educ.*, vol. 51, no. 1, pp. 24–37, 2003.

[28] R. Gurley, "Student perception of the effectiveness of SmartMusic as a practice and assessment tool on middle school and high school band students," Texas Tech University, 2012.

[29] Y. Kafai, "Constructionism," in *Cambridge Handbook of the Learning Sciences*, K. Sawyer, Ed. Cambridge, MA: Cambridge University Press, 2006, pp. 35–46.

[30] L. Hetland, L. Winner, S. Veenema, and K. Sheridan, *Studio Thinking: The Real Benefits of Arts Education*. New York, NY: Teachers College Press, 2007.

[31] D. W. Shaffer and M. Resnick, "' Thick' Authenticity: New Media and Authentic Learning.," *J. Interact. Learn. Res.*, vol. 10, no. 2, pp. 195–215, 1999.

[32] H.-S. Lee and N. Butler, "Making authentic science accessible to students - International Journal of Science Education," *Int. J. Sci. Educ.*, vol. 25, no. 8, pp. 923–948, 2003.

[33] O. Astrachan and A. Briggs, "The CS Principles Project," *ACM Inroads*, vol. 3, no. 2, pp. 38–42, Jun. 2012.

[34] T. Magnusson, "ixi lang: A SuperCollider Parasite for Live Coding," in *SuperCollider Symposium 2010*, Berlin, 2010.

[35] J. Freeman and A. V. Troyer, "Collaborative Textual Improvisation in a Laptop Ensemble," *Comput. Music J.*, vol. 35, no. 2, pp. 8–21, May 2011.

[36] A. Mahadevan, J. Freeman, B. Magerko, and J. C. Martinez, "EarSketch: Teaching computational music remixing in an online Web Audio based learning environment," in *Proceedings of the 1st Annual Web Audio Conference*, Paris, 2015.

[37] D. Bau, "Droplet, a blocks-based editor for text code," *J. Comput. Sci. Coll.*, vol. 30, no. 6, pp. 138–144, 2015.

[38] J. Freeman and B. Magerko, "Iterative composition, coding and pedagogy: A case study in live coding with EarSketch," *J. Music Technol. Educ.*, vol. 9, no. 1, 2016.

[39] J. Freeman, "Survey of Music Technology," *Coursera*, 2015. [Online]. Available: https://www.coursera.org/learn/music-technology. [Accessed: 25-Mar-2016].

[40] E. Wiebe, L. Williams, K. Yang, and C. Miller, "Computer science attitude survey," *Comput. Sci.*, vol. 14, no. 25, 2003.

[41] G. Knezek and R. Christensen, "Validating the Computer Attitude Questionnaire (CAQ).," 1996. [Online]. Available: http://files.eric.ed.gov/fulltext/ED398243.pdf.

[42] E. A. Carroll, C. Latulipe, R. Fung, and M. Terry, "Creativity factor evaluation: towards a standardized survey metric for creativity support," in *Proceedings of the seventh ACM conference on Creativity and cognition*, 2009, pp. 127–136.

[43] T. M. Amabile, "Within you, without you: The social psychology of creativity, and beyond," *Theor. Creat.*, vol. 4, pp. 61–91, 1990.

[44] R. E. Mayer, "22 Fifty Years of Creativity Research," *Handb. Creat.*, vol. 449, 1999.

[45] B. Magerko, J. Freeman, T. McKlin, M. Reilly, E. Livingston, S. McCoid, and A. Crews-Brown, "EarSketch: A STEAM-based Approach for Underrepresented Populations in High School Computer Science Education," *ACM Trans. Comput. Educ.*, in press 2016.

[46] D. Llewellyn, M. Usselman, D. Edwards, R. Moore, and P. Mital, "Analyzing K-12 Education as a Complex System," in *Proceedings of the ASEE 2013 Annual Conference*, 2013.

[47] L. McCarthy, "p5.js," 2016. [Online]. Available: http://p5js.org. [Accessed: 26-Mar-2016].

[48] C. Howe, "Analog Synthesizers in the Classroom," Georgia Institute of Technology, Atlanta, GA, 2014.

[49] L. Buechley, M. Eisenberg, J. Catchen, and A. Crockett, "The LilyPad Arduino: using computational textiles to investigate engagement, aesthetics, and diversity in computer science education," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, New York, NY, USA, 2008, pp. 423–432.

THE SELFEAR PROJECT: A MOBILE APPLICATION FOR LOW-COST PINNA-RELATED TRANSEFR FUNCTION ACQUISITION

Michele Geronazzo
Dept. of Neurological and Movement Sciences
University of Verona
michele.geronazzo@univr.it

Jacopo Fantin, Giacomo Sorato, Guido Baldovino, Federico Avanzini
Dept. of Information Engineering
University of Padova
Correspondence should be addressed to
avanzini@dei.unipd.it

ABSTRACT

Virtual and augmented reality are expected to become more and more influential even in everyday life in the next future; the role of spatial audio technologies over headphones will be pivotal for application scenarios which involve mobility. This paper faces the issue of head-related transfer function (HRTF) acquisition with low-cost mobile devices, affordable to anybody, anywhere and possibly in a faster way than the existing measurement methods. In particular, the proposed solution, called the SelfEar project, focuses on capturing individual spectral features included in the pinna-related transfer function (PRTF) guiding the user in collecting non-anechoic HRTFs through a self-adjustable procedure. Acoustic data are acquired by an audio augmented reality headset which embedded a pair of microphones at listener ear-canals. The proposed measurement session captures PRTF spectral features of KEMAR mannequin which are consistent to those of anechoic measurement procedures. In both cases, the results would be dependent on microphone placement, minimizing subject movements which would occur with human users. Considering quality and variability of the reported results as well as the resources needed, the SelfEar project proposes an attractive solution for low-cost HRTF personalization procedure.

1. INTRODUCTION

Binaural audio technologies have the aim of reproducing sounds in the most natural way, as if listeners were surrounded by realistic virtual sound-sources. This audio technology originated in late 19th century experiments [1], and it finds its roots in the recording of sounds through a "dummy head" that simulates the characteristics of the listener's head and incorporates two microphonic capsules inside the auditory ducts, emulating eardrums membranes [2]. Binaural audio could provide us with a 360 degrees listening experience, placing the virtual sound sources in defined points thanks to which our brain succeeds in perceiving the spatial qualities of source and environment. It obtains its maximum efficiency through headphones reproduction, which keeps signal characteristics intact, without environmental reflections and reverberations. The rendering of virtual acoustic scenarios involves binaural room impulse responses (BRIR) that can be defined in two main components: the first one is connected to the environmental characteristics contained in the room impulse response (RIR), and the other one is related to the anthropometric characteristics of the listener, i.e. head-related impulse response (HRIR) [2]. All these impulse responses (IRs) have their counterparts in the frequency domain, formally their Fourier transforms: binaural room transfer function (BRTF), room transfer function (RTF), and head-related transfer function (HRTF). In particular, HRTFs describe a linear time-invariant filter where the acoustic filtering to which head, torso and ear of a subject concur is defined.

The ground-truth HRTF acoustic measurement offers an impulse response that has high-quality subject-related information and high-precision. However, professional HRTFs acquirement process requires time resources and expensive equipments that are rarely available for real applications. A more affordable procedure could discard some individual features to obtain a cheaper HRTF representation which still gives accurate psyco-acoustic information [3]. The HRTF acquirement process in a domestic environment is a challenging issue; recent trends are supported by low-cost devices for acquisition of 3D mesh images [4] and algorithms for HRTF modeling and customization [5]. These solutions unfortunately lack robust individual cues for external ear acoustics due to the fine anthropometric structure of the pinna. This information is collected in the so called pinna-related transfer function (PRTF) [6] which is also very difficult to model in numerical simulations [7, 8]. PRTFs contain salient localization cues for elevation perception (see [9] for a review), requiring an accurate representation in order to provide vertical dimension in binaural audio technologies.

This paper highlights the issue of costs reduction in the HRTF acquirement process, with particular focus on PRTF extrapolation for the mobile audio augmented reality (mAAR) system. This system involve headphones, provided with embedded external microphones for binaural capture of multiple-channel audio stream from the environment, as well as algorithms for binaural audio reproduction. An attractive idea is to use embedded micro-

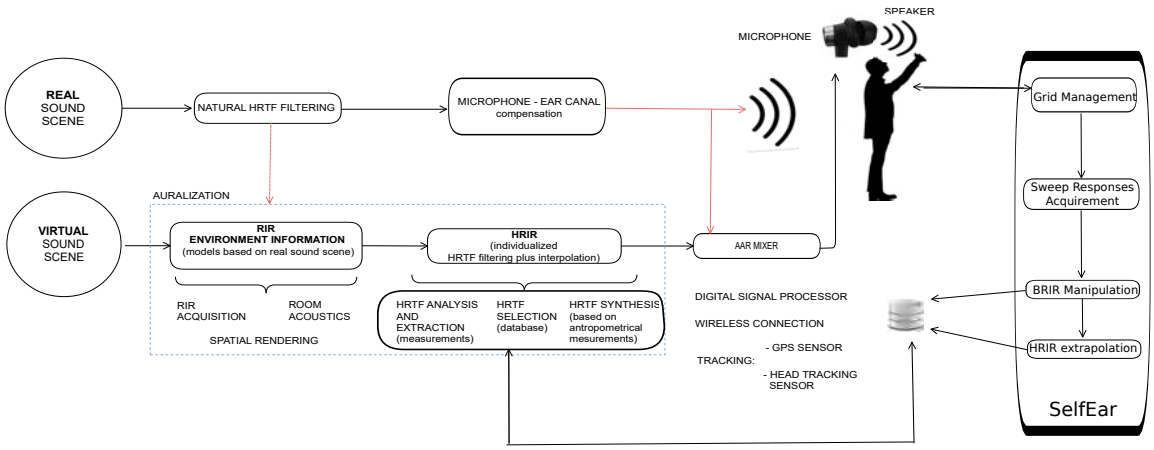


Figure 1: Schematic view of the SelfEar project in mAAR contexts.

phones in order to acquire HRTFs everywhere from sound stimuli played back by mobile device's speakers; the SelfEar project has the purpose of developing the signal processing algorithms and interaction with the device in order to obtain a self-adjust procedure. Few studies have been conducted aiming to access the HRTF consistency in a non-anechoic environment for the acoustic contribution in mid-sagittal planes [10] which are relevant for individual spectral content introduced in PRTFs. The compromise on costs and portability unavoidably leads to mainly two different issues. Firstly, the mobile acquirement process implicates surrounding environment influences such as frequency coloration and phase shifts. Secondly, employing mobile device's speakers as sound source and consumer binaural microphones for the acquisition brings to less accurate recordings with respect to professional equipement.

In this paper we presented a series of measurements conducted in a silent booth on a KEMAR dummy head [11]. Our final goal was to compare responses obtained using the SelfEar system with those from professional equipment. In particular: Sec. 2 contains the description of a mobile audio augmented reality system and criteria for virtual sound externalization; in Sec. 3 the SelfEar project is presented. Section 4 describes acoustic measurements on a dummy head in non-anechoic environment. Finally, results are discussed in Sec. 5, and Sec. 6 concludes the proposed preliminary evaluation with promising research directions.

2. MOBILE AUDIO AUGMENTED REALITY

In a mAAR system (see fig.1), the listener might be able to enjoy a mix of real and virtual sound sources. The real sound sources are captured by headset microphones after natural acoustic filtering by the listener. A compensation filter considers errors introduced by different headphones and microphones positions compared to the unblocked entry point of the auditory channel resembling natural listener condition. The rendering of virtual sources needs a dynamic and parametric auralization process in order to create a perfect superposition with reality. Auralization employs BRIR, whose rendering must be coherently connected to the real surrounding environment in which the subject is immersed. The cascade of RIRs and HRIRs should be personalized according to environment [12] and the listener [3]. Digital signal processing (DSP) algorithms implement corrective filters that compensate microphones, speakers and their interactions, taking into account psychoacoustic effects and artifacts that may be caused by wearing the earphone with respect to normal hearing conditions without headset.

Producing realistic virtual and augmented acoustic scenarios over headphones with particular attention to space properties and externalization issues remains one major challenge due to the interconnections of the above mentioned components of a mAAR system. Challenges and criteria for reality driven externalization can be summarized in four categories [13]:

- *ergonomic delivery system*: the ideal headphones should be acoustically transparent which means listeners are not aware of the sound emitted by transducers [14]. Low invasiveness of headphones cups are essential for such purpose [15].

- *tracking*: head movements in listening produces reliable dynamic interaural cues [16]; tracking listener position in the environment allows recognition of acoustic interaction and a common spatial representation between real and virtual scene;

- *room acoustics knowledge*: spatial impression and perception of the acoustic space involve the knowledge of real world early reflection and reverberation [17]; this information concurs to the availability of realistic spatial impression [18];

- *individual spectral cues*: head and pinna individually filter the incoming sound to listener ears; moreover individual correction must be considered for acoustic coupling between headphones and external ear [19].

3. THE SELFEAR PROJECT

3.1 Overview of the system

SelfEar is a mobile application designed to be executed on the Android platform in order to obtain user's personal HRIRs from a sound stimulus played by the mobile device. The phone/tablet must be held with the stretched arm and moved on the subject's median plane stopping at specific arm's elevation angle. The in-ear microphones capture the audio coming from the loudspeaker device, thus recording the position-, listener- and environment- specific BRIR, i.e. an acoustic self-portrait. The data collected through this application can be later employed in order to finally obtain an individualized HRIR. After post-processing procedures that compensate acoustic effect of acquiring conditions and playback device, individualized HRTFs can directly support spatial audio rendering and research framework [20]. Depending on the complexity of virtual scenarios, real-time HRTF synthesis is possible on mobile platform today. A promising technique involves HRTF selection through acoustic parameter extracted with SelfEar: the procedure selects the subject's best HRTF approximation based on existing HRTF databases (for instance CIPIC database [21]). [1]

3.2 Source manager

The spatial grid management system of SelfEar guides the user through the BRIR acquirement process defining a self-adjusted procedure depicted in Fig. 2. In the following, we describe each step, starting from the application launch to the session end, resulting in a set of individual BRIRs.

In the launching view of the SelfEar application, the user is asked to select the device's speakers position that may be on the top, front, bottom or back side of the device. This choice will have an effect on the device orientation during the sound stimulus playback in order to maximize speakers performance due to their directivity. The user can then press the "'Start"' button to begin the BRIR acquirement procedure; its steps follow this logical flow:

1. *Target reaching*: the current device elevation in the mid-sagittal plane appears on the scree above the target elevation (see the screenshot on the bottom right of Fig. 2). SelfEar computes data coming from the device's accelerometer on the three Cartesian axes, $a_{x,y,z}$, to calculate the current elevation on the horizon, ϕ_i, with the following formula:

$$\phi_i = \arctan\left(\frac{\pm a_y}{|a_z|}\right)$$

in case the speakers are located in the top or bottom side; whereas with the formula:

$$\phi_i = \arctan\left(\frac{\pm a_z}{|a_y|}\right)$$

in case in the front or back side. The numerator has the sign equals to:

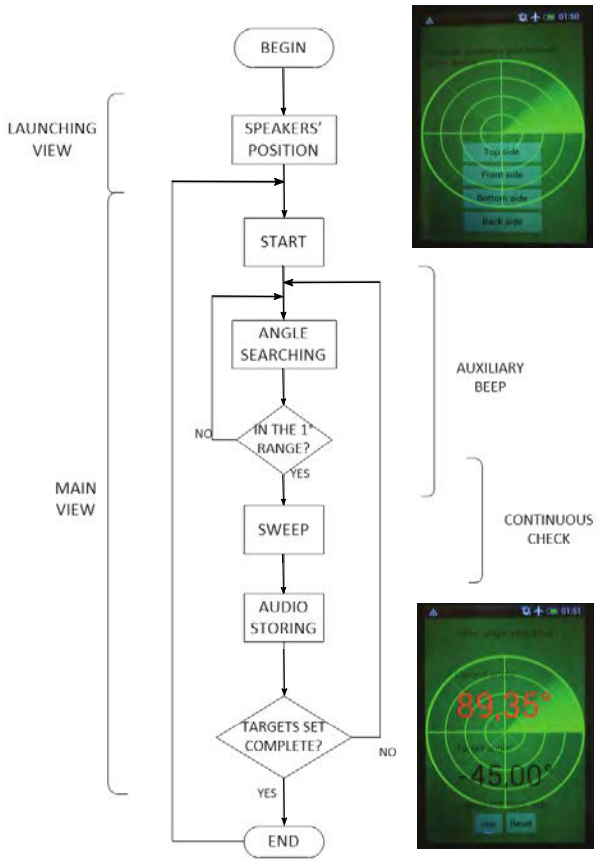


Figure 2: Block diagram of SelfEar procedure for BRIR acquisition in the median plane. Screenshots of the two application views are also reported.

+ for bottom- or back-sided speakers;
- for top- or front-sided speakers.

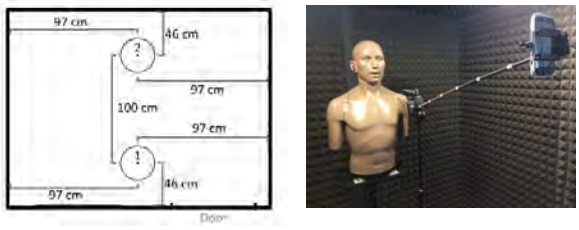
Target elevations sequence spans in ascending order among $[-40°, 40°]$ angles of the CIPIC HRTF database with equal spacing of $5.625°$. An auxiliary beep signal sonifies the distance between the actual and the target position supporting the elevation pointing procedure, which would be particularly useful in case the display is not visible due to the speaker's position (e.g. in the back side). The pause between one beep and another is directly proportional to the difference between the current measured angle, ϕ_i and the target, $\widehat{\phi}_i$, as shown in the following equation:

$$pause_i = \left|\phi_i - \widehat{\phi}_i\right| \cdot k$$

where *i* is an instant when a single beep terminates its playback and *k* is a constant value that makes perceptible the pause. [2] The goal for this step is to approach the target elevation within a precision of

[1] A collection of several acoustic measurements conducted on 50 different subjects (more than 1200 measurements each), also including anthropometric information.

[2] The formula returns a value in milliseconds, which would result in a too short pause to be heard without a constant multiplier. For the proposed implementation, we chose $k = 5$ with informal tests.



(a) (b)

Figure 3: Measurement setup. (a) Source and receiver positions in the SSP. (b) SelfEar measument setup with selfie stick incorporated.

$\pm 1°$. This step can be interrupted and resumed upon request by the user.

2. *Position check*: once ϕ_i enters the valid range, a stability timer of 2 seconds starts; should the number of times the user exits a range of $\pm 2°$ from the target reach three before the timer ends, the procedure goes back to the end of step 1.

3. *Sweep playback*: after the stability timer ends, the sound stimulus will be played from the device's speakers; should the user exit the $\pm 2°$ range just once during the sweep playback, the searching procedure for $\widehat{\phi}_i$ is reset.

4. *BRIR storing*: once a sweep successfully terminates, the recorded audio is locally stored together with the elevation angle it refers to; the procedure then returns to step 1 with next target elevation in the sequence.

5. *End of session*: a session ends when elevations in the targets set are successfully reached.

4. ACOUSTIC MEASUREMENTS

Two measurement sessions were performed in a non-anechoic environment using a dummy head in order to minimize errors due to subject movement. We focused on the frontal direction $\phi = 0$ [6, 22] which is the spatial direction with highly significant PRTF spectral characteristics: the two main resonances (P1: omnidirectional mode, and P2: horizontal mode) and the three prominent notches (N1-3 corresponding to pinna reflections). Accordingly, we provided a detailed analysis of the acquired acoustic signals with different measurement setups, also reporting a qualitative evaluation of the SelfEar application for a set of HRIRs in the frontal mid-sagittal plane.

4.1 Setup

Facility and Equipment - All the measurement and experimental sessions were conducted inside a Sound Station Pro 45 (SSP), a 2×2 m silent booth with a maximum acoustic isolation of 45 dB.

Figure 3a shows the spatial setup of each experiment measurement in the SSP, identifying two positions: position #1 relative to the source, while position #2 to the receiver.

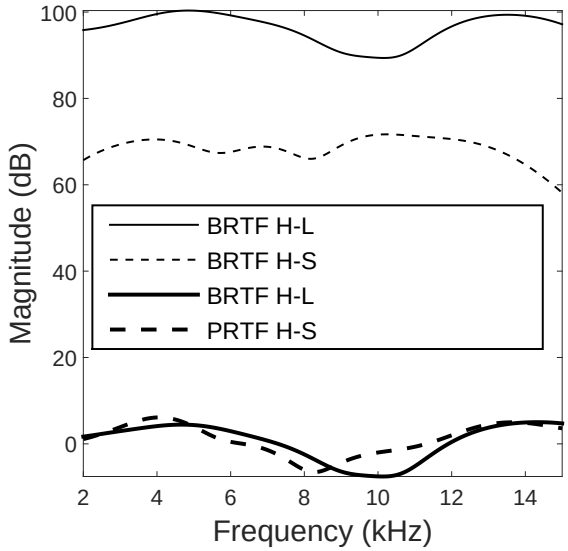


Figure 4: Magnitude comparison (in dB SPL) of BRTFs (thick lines) and relative PRTFs (thin lines) obtained using: receiver - the right headset microphone (H), source - the smartphone loudspeaker (S, dashed lines) and the Genelec loudspeaker (L, continuous lines).

Two types of playback device have been used in the experiments (acronyms also defined):

- *L* : a Genelec 8030A loudspeaker which has been calibrated to have an adequate SNR with a test tone at 500 Hz with 94 dB SPL;
- *S* : a HTC Desire C smartphone supported by a self-produced boom arm with a selfie stick incorporated; [3] in this case the maximum SPL reached is 51 dB at the reference frequency of 500 Hz.

Two type of receivers were also used in all the measurements (acronyms also defined):

- *H* : a pair of Roland CS-10EM in-ear headphones with embedded microphones;
- *K* : professional G.R.A.S microphones embedded in the head&torso simulator KEMAR; in the proposed setup, the right ear was equipped with ear canal simulator while the left ear not.

In all experiments, the center of sound source and receiver were placed at the same height. The source signal was a one second logarithmic sine sweep signal that comprises all the audible frequencies, from 20 Hz to 20 kHz, uniformly. The acoustic signals were recorded with the free software Audacity with a Motu 896 mk 3 audio interface and the processing was accomplished in Matlab (version 8.4).

[3] Since the 1-m selfie-stick is longer than an average arm of the user, we assume that PRTF spectral details for elevation perception are invariant with distance [23].

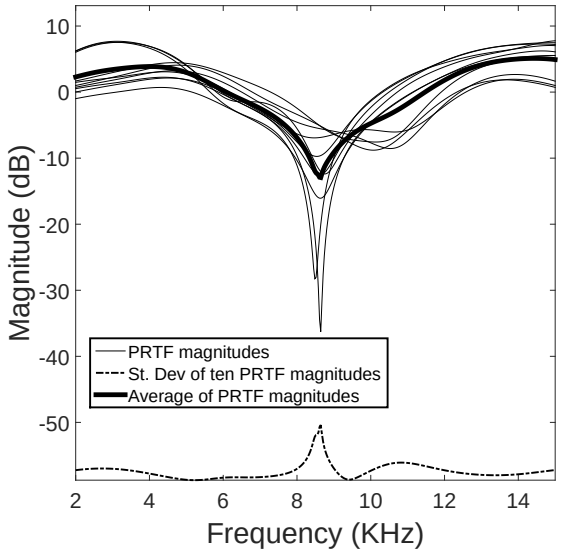


Figure 5: PRTF magnitudes in ten repositioning of the headset in the right ear canal of the KEMAR mannequin. Thick line represents the average magnitude. The standard deviation is shifted by -60 dB for convenience.

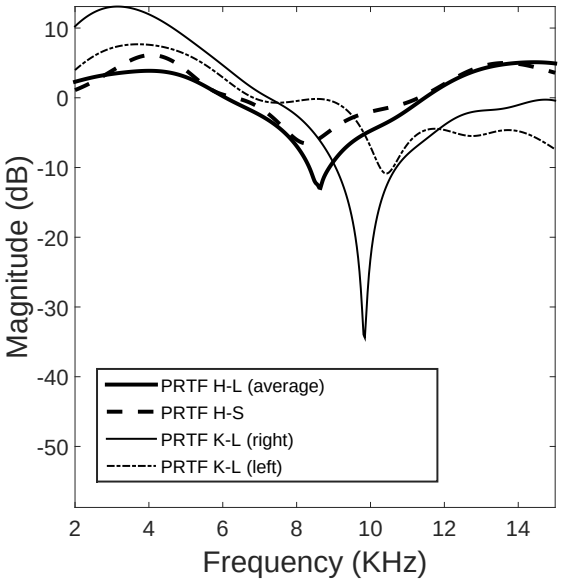


Figure 6: PRTF magnitude comparison: a) average PRTF from Fig.5; b) source: smartphone - receiver: headphone microphones; c) source: Genelec loudspeaker - receiver: KEMAR microphone in the right ear with ear canal; d) source: Genelec loudspeaker - receiver: KEMAR microphone in the left ear without ear canal simulator.

Calibration: diffuse-field measurement - A self-produced structure was used for diffuse-field measurements in order to acquire environmental- and setup- specific acoustical features. It consists of two pieces of iron wire that fall from the booth ceiling at a distance of 17.4 cm apart, corresponding to the same distance of KEMAR microphones.

We acquired diffuse-field measurements for all pairs of source and receiver, leading to a total of four measurements.

4.2 Acoustic data

Measurement session one - In this session, the Genelec loudspeaker and the KEMAR were placed inside the SSP , respectively in positions #1 and #2 of Fig. 3a. In the first step, right and left ear response of KEMAR were measured thus obtaining an *at the eardrum* measurement for the right ear and a *blocked ear canal* measurement for the left ear. The second step involved the headset inserted in the right ear canal; we conducted ten measurements related to different earphones placements in order to analyze measurement variability introduced by microphone position.

Measurement session two - In this session, the selfie-stick structure held the smartphones which was placed inside the SSP in position #1 of Fig. 3a; on the other hand, the KEMAR wearing the right headphone was placed in position #2 of Fig. 3a. The self-stick structure kept the smartphone at the distance of one meter from the KEMAR and allowed a fine angular adjustment. Measurements spanned 15 angles between $-40°$ and $+40°$ on the median plane. Finally, we obtained two sets of 15 measurements for the left KEMAR ear (without headphones) and the right headphone microphone.

4.3 Analysis

For each measurement, the onset detection was computed applying a cross-correlation function with the original sweep signal and the BRIR was then extracted deconvolving sweep responses with the same sweep. Late reflections caused by the SSP and the presence of the equipment in the SSP were removed subtracting the corresponding diffuse-field responses from BRIRs. This processing ensured the acquirement of HRTFs. Accordingly, PRTFs were obtained by windowing each impulse response with a 1 -ms hanning window (48 samples) temporally-centered on the maximum peak and normalized on the maximum value in amplitude [6]. All of normalized PRTF were then band-pass filtered between 2 kHz and 15 kHz, ensuring the extraction of salient peaks and notches caused by pinna acoustics.

Figure 4 shown the comparison between the magnitudes in dB SPL of the BRIR extracted from the measurements using as source (i) the Genelec loudspeaker, (ii) the smartphone loudspeaker, and the headset on the right KEMAR ear as receiver. It has to be noted that the sound pressure levels of the two loudspeakers differed from 30 dB SPL on average denoting a low signal-to-noise ratio while using smartphone loudspeaker. The same figure also depicts the two corresponding normalized PRTFs in order to assess the diffuse-field effects on the results. For smartphone measurements, the contribution of the diffuse-field compensation is clearly visible due to non-negligible acoustic contribution of the low-cost loudspeaker.

In Fig. 5, the dB magnitude of PRTFs of ten repositionings and their average are reported. The standard deviation is also reported in order to analyze variability in the mea-

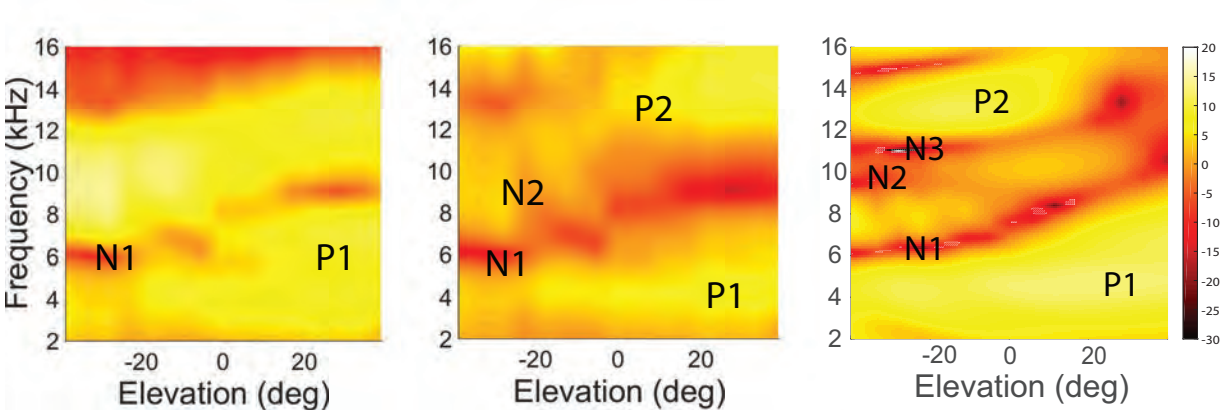


Figure 7: PRTFs in the median plane. (a) SelfEar acquisition - no compensation; (b) SelfEar acquisition - with diffuse-field compensation; (c) CIPIC KEMAR, Subject 165 - with free-field compensation. Plots also contain labels for the main peaks (P1-2) and notches (N1-3), where present.

surements introduced by headphone/microphone position. The maximum variability occurred in proximity of salient PRTF notches at 9 and 11 kHz which exhibited high sensitivities to topological changes between headphones and ear structure [8].

The main quantitative evaluation was performed in the frontal source position, $\phi = 0$, comparing the normalized PRTFs in different conditions. Figure 6 showed comparisons among PRTF magnitudes of measurements acquired with and without headset involving both Genelec and smartphone loudspeaker.

For this four PRTFs, the average spectral distortion (SD) error has been calculated [9] pairwise in the frequencies of interest 2 kHz $\leq$ f $\leq$ 15 kHz (value are showed in Table 1). These comparisons lead to several considerations:

- Pinna acoustics, K-L$_{right}$ vs. K-L$_{left}$: different ear shapes (right vs. left), and the ear canal acoustics (right with ear canal simulator and left with the blocked ear canal) differed remarkably; all comparisons between the 3^{rd} and 4^{th} column reflected these differences;
- Loudspeakers, H-S$_{right}$ vs. H-L$_{right}$: different loudspeakers introduced negligible spectral distortion in the proposed setup ($<$ 2 dB);
- SelfEar procedure, H-S$_{right}$ vs. K-L$_{left}$: difference between SelfEar acquisition of PRTFs and traditional measurement setup introduced the lower SD error in the available set (removing the control comparison on loudspeakers);

PRTF	H-L	H-S	K-L(right)	K-L(left)
H-L	0	1.79	6.92	5.25
H-S		0	7.35	4.64
K-L(right)			0	5.47
K-L(left)				0

Table 1: Spectral distortion among PRTFs of Figure 6. All values are in dB.

Figure 7 allows a visual comparison from the results obtained using SelfEar acquirement procedure on the considered elevation angles (with and without diffuse-field compensation), and the CIPIC measurements on the same angles range for Subject 165. The data were interpolated in order to have a smooth spatial transition.

5. DISCUSSION

From Christensen *et al.* [24] it is already known that the receiver position and its displacement from the ideal HRTF measurement point, i.e. at the entrance of the ear canal, highly influence HRTF directivity patterns for frequencies higher than $3-4$ kHz. Our work is in agreement with their measurements showing a shift of notch central frequencies up to 2 kHz with very high variability in magnitude among different microphone placements (see standard deviation of Fig. 5) and a maximum difference of 10 dB.

Shifts in peak/notch central frequencies are also visible in Fig. 6 due to topological differences between observation point, depending on microphone position, and acoustic scattering object, i.e. presence/absence of ear canal and differences between left and right ears. Spanning a wider range of frontal elevation positions allowed any measurement system to acquire relevant PRTF spectral features: in PRTFs from the CIPIC KEMAR (see labels in Fig. 7(c)), P1 has central frequency at 4 kHz and P2 at 13 kHz, moreover N1 moves from 6 to 9 kHz, N3 from 11.5 to 14 kHz with increase in elevation; finally, N2 stars from 10 kHz and progressively disappears once reaching the frontal direction.

SelfEar application is capable of acquire P1 and N1 effectively considering both diffuse-field compensated PRTFs or not compensated BRIRs. Since the environment had not negligible contribution, the visual comparison between Fig. 7(a) and (b) stresses the importance of being able to accurately extract PRTFs from BRIRs. In particular from Fig. 7(b), one can identify also P2 and a little presence of N2. However, N3 was completely absent suggesting an acoustic interference introduced by headphones in pinna

concha. Following the resonances-plus-reflections model for PRTFs [6, 9], we can speculate about the absence of concha reflections due to headphone presence; moreover, the volume of the concha was dramatically reduced in this condition, thus producing changes in resonant modes of the pinna structure [8]. Furthermore, SD value of comparison $H-S$ vs. $K-L_{left}$ is 4.64 dB which suggests a good reliability in performances comparable to the personalization method in [9] (SD values between 4 and 8 dB) and to the state-of-the art numerical HRTF simulations in [8] (SD values between 2.5 and 5.5 dB).

It is worthwhile to notice that notch and peak parameters, i.e. central frequency, gain, and bandwidth, can be directly computed from available PRTFs. These spectral features can be exploited in synthetic PRTF models and/or HRTF selection procedure following a mixed structural modeling approach [3]. Finally, there is nothing to prevent a direct usage of PRTFs extracted by SelfEar in binuaral audio rendering.

6. CONCLUSION AND FUTURE WORK

The SelfEar application allows low-cost HRTF acquisition in the frontal median plane capturing peculiar spectral cues of the listener's pinna, i.e. PRTF. The application take advantage of a AAR technological framework for mobile devices. Once properly compensated, extracted PRTFs are comparable in terms of salient acoustical features to those measured in anechoic chamber.

The proposed system was tested following a robust measurements setup without a human subject in a silent booth which is an acoustically treated environment. Thus, a robust procedure is require for PRTF capturing in domestic environments, statistically assessing the influence of noisy and random acoustic events, as well as subject movements during the acquisition. For such purpose, signal processing algorithms for event detection, noise cancellation and movement tracking are crucial in signal compensation and in pre- and post- processing stages.

A natural evolution of this application will take into account also sagittal planes, i.e. plane around listeners with azimuth $\neq 0$, with particular attention to frontal directions which are easily accessible with arm movements and are crucial for auditory displays such as sonified screens [25]. Optimized procedures will be studied in order to reduce the number of required source positions and to control mobile position and orientation with respect to user movements; the SelfEar application will implement computer vision algorithms able to track listener's head-pose in real-time with embedded camera and depth sensors.

In addition to HRTF acquisition functionality, we will include capabilities of full BRIR acquisition in SelfEar, storing RIR and HRIR responses separately in order to directly render mAAR scenarios coherently in real-time. Extrapolated RIR will parametrize computational room acoustic models for the purpose of dynamic auralization, such as image-source and raybeam-tracing modeling for the first reflections and statistical handling of late reverberation [12].

Finally, it is indisputable that psycho-acoustic evaluation with human subjects is necessary in order to confirm the reliability of the SelfEar application providing effective individualized HRIRs in rendering virtual sound sources.

Acknowledgments

This work was supported by the research project Personal Auditory Displays for Virtual Acoustics, University of Padova, under grant no. CPDA135702.

7. REFERENCES

[1] S. Paul, "Binaural Recording Technology: A Historical Review and Possible Future Developments," *Acta Acustica united with Acustica*, vol. 95, no. 5, pp. 767–788, Sep. 2009.

[2] J. Blauert, *Spatial Hearing: The Psychophysics of Human Sound Localization*. Cambridge, MA, USA: MIT Press, 1983.

[3] M. Geronazzo, S. Spagnol, and F. Avanzini, "Mixed Structural Modeling of Head-Related Transfer Functions for Customized Binaural Audio Delivery," in *Proc. 18th Int. Conf. Digital Signal Process. (DSP 2013)*, Santorini, Greece, Jul. 2013, pp. 1–8.

[4] H. Gamper, M. R. P. Thomas, and I. J. Tashev, "Anthropometric Parameterisation of a Spherical Scatterer ITD Model with Arbitrary Ear Angles," in *2015 IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA)*, Oct. 2015, pp. 1–5.

[5] S. Spagnol, M. Geronazzo, D. Rocchesso, and F. Avanzini, "Extraction of Pinna Features for Customized Binaural Audio Delivery on Mobile Devices," in *Proc. 11th Int. Conf. on Advances in Mobile Computing & Multimedia (MoMM13)*, Vienna, Austria, Dec. 2013, pp. 514–517.

[6] M. Geronazzo, S. Spagnol, and F. Avanzini, "Estimation and Modeling of Pinna-Related Transfer Functions," in *Proc. of the 13th Int. Conf. on Digital Audio Effects (DAFx-10)*, Graz, Austria, Sep. 2010, pp. 431–438.

[7] H. Ziegelwanger, P. Majdak, and W. Kreuzer, "Numerical Calculation of Listener-specific Head-related Transfer Functions and Sound Localization: Microphone Model and Mesh Discretization," *J. Acoust. Soc. Am.*, vol. 138, no. 1, pp. 208–222, Jul. 2015.

[8] S. Prepelită, M. Geronazzo, F. Avanzini, and L. Savioja, "Influence of Voxelization on Finite Difference Time Domain Simulations of Head-Related Transfer Functions," *J. Acoust. Soc. Am.*, vol. 139, no. 5, pp. 2489–2504, May 2016.

[9] S. Spagnol, M. Geronazzo, and F. Avanzini, "On the Relation between Pinna Reflection Patterns and Head-Related Transfer Function Features," *IEEE Trans. Audio, Speech, Lang. Process.*, vol. 21, no. 3, pp. 508–519, Mar. 2013.

[10] A. Ihlefeld and B. Shinn-Cunningham, "Disentangling the Effects of Spatial Cues on Selection and Formation of Auditory Objects," *J. Acoust. Soc. Am.*, vol. 124, no. 4, pp. 2224–2235, 2008.

[11] W. G. Gardner and K. D. Martin, "HRTF Measurements of a KEMAR," *J. Acoust. Soc. Am.*, vol. 97, no. 6, pp. 3907–3908, Jun. 1995.

[12] L. Savioja and U. P. Svensson, "Overview of Geometrical Room Acoustic Modeling Techniques," *J. Acoust. Soc. Am.*, vol. 138, no. 2, pp. 708–730, Aug. 2015.

[13] J. Loomis, R. Klatzky, and R. Golledge, "Auditory Distance Perception in Real, Virtual and Mixed Environments," in *Mixed Reality: Merging Real and Virtual Worlds*, Y. Ohta and H. Tamura, Eds. Springer, 1999.

[14] J. Ramo and V. Valimaki, "Digital Augmented Reality Audio Headset," *J. of Electrical and Computer Engineering*, vol. 2012, p. e457374, Oct. 2012.

[15] R. W. Lindeman, H. Noma, and P. G. d. Barros, "An Empirical Study of Hear-Through Augmented Reality: Using Bone Conduction to Deliver Spatialized Audio," in *2008 IEEE Virtual Reality Conference*, Mar. 2008, pp. 35–42, 00013.

[16] W. O. Brimijoin, A. W. Boyd, and M. A. Akeroyd, "The Contribution of Head Movement to the Externalization and Internalization of Sounds," *PLoS ONE*, vol. 8, no. 12, p. e83068, Dec. 2013.

[17] N. Sakamoto, T. Gotoh, and Y. Kimura, "On -Out-of-Head Localization- in Headphone Listening," *J. of the Audio Eng. Soc.*, vol. 24, no. 9, pp. 710–716, Nov. 1976.

[18] J. S. Bradley and G. A. Soulodre, "Objective Measures of Listener Envelopment," *J. Acoust. Soc. Am.*, vol. 98, no. 5, pp. 2590–2597, Nov. 1995.

[19] F. L. Wightman and D. J. Kistler, "Headphone Simulation of Free-Field Listening. II: Psychophysical validation," *J. Acoust. Soc. Am.*, vol. 85, no. 2, pp. 868–878, 1989.

[20] M. Geronazzo, S. Spagnol, and F. Avanzini, "A Modular Framework for the Analysis and Synthesis of Head-Related Transfer Functions," in *Proc. 134th Conv. Audio Eng. Society*, Rome, Italy, May 2013.

[21] V. R. Algazi, R. O. Duda, D. M. Thompson, and C. Avendano, "The CIPIC HRTF Database," in *Proc. IEEE Work. Appl. Signal Process., Audio, Acoust.*, New Paltz, New York, USA, Oct. 2001, pp. 1–4.

[22] F. Asano, Y. Suzuki, and T. Sone, "Role of Spectral Cues in Median Plane Localization," *J. Acoust. Soc. Am.*, vol. 88, no. 1, pp. 159–168, 1990.

[23] D. S. Brungart and W. M. Rabinowitz, "Auditory localization of nearby sources. Head-related transfer functions," *J. Acoust. Soc. Am.*, vol. 106, no. 3, pp. 1465–1479, 1999.

[24] F. Christensen, P. F. Hoffmann, and D. Hammershøi, "Measuring Directional Characteristics of In-Ear Recording Devices," in *In Proc. Audio Eng. Soc. Con. 134*. Audio Engineering Society, May 2013.

[25] A. Walker and S. Brewster, "Spatial Audio in Small Screen Device Displays," *Pers. Technol.*, vol. 4, no. 2, pp. 144–154, Jun. 2000.

PITCH CONTOUR SEGMENTATION FOR COMPUTER-AIDED JINGJU SINGING TRAINING

Rong Gong, Yile Yang, Xavier Serra
Music Technology Group
Universitat Pompeu Fabra
Barcelona, Spain
{rong.gong,yile.yang,xavier.serra}@upf.edu

ABSTRACT

Imitation is the main approach of jingju (also known as Beijing opera) singing training through its inheritance of nearly 200 years. Students learn singing by receiving auditory and gestural feedback cues. The aim of computer-aided training is to visually reveal the student's intonation problem by representing the pitch contour on segment-level. In this paper, we propose a technique for this purpose. Pitch contour of each musical note is segmented automatically by a melodic transcription algorithm incorporated with a genre-specific musicological model of jingju singing: bigram note transition probabilities defining the probabilities of a transition from one note to another. A finer segmentation which takes into account the high variability of steady segments in jingju context enables us to analyze the subtle details of the intonation by subdividing the note's pitch contour into a chain of three basic vocal expression segments: steady, transitory and vibrato. The evaluation suggests that this technique outperforms the state of the art methods for jingju singing. The web prototype implementation of these techniques offers a great potential for both in-class learning and self-learning.

1. INTRODUCTION

Jingju is a traditional Chinese theater which combines music, vocal performance, mime, dance, and acrobatics. It arose in the late 18th century and became fully developed and recognized by the mid-19th century. The form was extremely popular in the Qing dynasty court and has come to be regarded as one of the cultural treasures of China [1].

Singing, one of the most basic means of performing in jingju, is fundamentally different from the music system in the West. There are currently four main role-types in jingju: *sheng*, *dan*, *jing*, *chou*. Different role-types have widely different singing styles. For example, the role of *laosheng* (elderly man) mainly uses the real voice, whereas the role of *dan* (woman) uses mainly falsetto [2].

Imitation is the main method of jingju professional singing training nowadays. During the training class, the teacher has absolute authority and his/her singing is seen as the standard to which the student's imitation should be as close as possible. Students do not have a big space for developing their own singing style until graduation.

The basic unit of jingju arias is couplets. Each couplet consists of two lines: opening line and closing line. The teacher sings firstly one line of a couplet which usually contains 7 or 10 syllables, then the student imitates. Finally, the teacher will give comments from the perspectives of intonation, rhythm, timbre, loudness and phonation. The assessing process is conducted by comparing the student's singing performance with the teacher's version on multilevel time scales which includes line-level, syllabic level, note-level and segment-level. The state of the art computer-aided singing training methods [3–6] devote most of their efforts on line-level and note-level intonation assessment because according to the research on Western music [7], the intonation is the predominant perceptual dimension for the musicians to judge the goodness of the singing performance. Similarly, intonation accuracy is also the very basic assessment dimension for jingju singing training, such that it is the research object of this work. Other assessment dimensions include phonation, rhythm, loudness and timbre.

In jingju singing assessment, the teacher would pay much attention on various vocal expressions in terms of intonation which are represented as different pitch contour segments, such as vibrato and transitory segments. Different vocal expressions will be dealt with on their characteristics in the singing assessment process. For the steady segment, we mainly evaluate its length and average pitch. For the transitory segment, we evaluate its slope, starting and ending pitches. For the vibrato, its rate, extent, length and average frequency will be evaluated.

The preliminary step of this task is to transcribe the singing performance to pitch contours of musical notes. This problem can be solved by melodic transcription algorithms [8]. Then we perform the segmentation of note's pitch contour in order to perform the comparison with a small granularity. We concentrate on three main pitch contour segment categories: steady, transitory and vibrato segments. We limit our discussion in this paper to these because they are three of the most important vocal expressions that determine the correctness of the intonation of jingju singing.

This paper is organized as follows: Section 2 presents an overview about the related techniques. Section 3 de-

scribes the jingju singing specific approach of pitch contour segmentation. Section 4 presents the dataset and the evaluation of these techniques. Section 5 introduces a web prototype for jingju singing training and Section 6 draws the conclusion of this work.

2. BACKGROUND

Mauch [9] defined melodic transcription as symbolizing the pitch (or fundamental frequency, f_0) track to the stable segments containing pitch and duration features, that is to say, the algorithms compute a time-pitch representation which needs to be further processed in order to detect note events with a discrete pitch value, an onset time and an offset time [8]. [9] introduced a melodic transcription algorithm whose first stage - pYIN is a modification of YIN algorithm [10] which outputs multiple pitch candidates together with their probabilities and then finds a smooth path through the candidates by using a HMM. The note tracking stage is performed as Viterbi-decoding of an HMM which simplified the approach of Viitaniemi [11] by replacing the observation probability distribution with a simple Gaussian Distribution and removing the duration model. In addition, it doesn't deal with notes quantised to the integer MIDI scale which allows a more fine-grained analysis and makes this simplified approach more prone to the singing performance evaluation.

Recently, the increasing interest of the MIR community in the application of music analysis techniques to non-Western music has drew attention to the fact that different musical genres necessitate different analysis techniques [12]. The incorporation of genre-specific knowledge such as rhythmic structure and tonality improves the transcription accuracy [13]. Musicological model of [11] which includes key signature probabilities and bigram note transition probabilities is also the implementation of this concept. However, because occurrence probabilities of different note values given the key have not been estimated for jingju singing, the genre-specific tonality knowledge cannot be directly used.

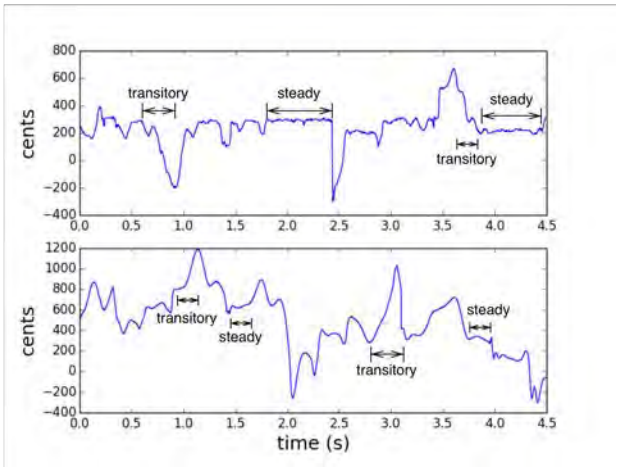


Figure 1. Pitch contour examples of Hindustani vocal (upper) and jingju vocal (lower). 0 cents is 261.6 Hz (C4).

The related problem of pitch contour segmentation has been studied for Hindustani vocal music, [14, 15] detect the steady segment in Hindustani vocal music by tracking the predominant peaks from the pitch histogram (pitch class distribution), then fit a 3 degrees polynomial to the remaining transitory segments. [16] obtains a canonical representation of the pitch contour in terms of straight lines and critical points of inflection, then recognizes two vocal expressions - *andolan* and *meend* by using their templates. In contrast to Hindustani music, the intonation of jingju singing shows more variability in steady segments (Figure 1). However, the transitory segment of jingju singing doesn't contain complicated ornaments such as undulating glide [17] and is thus less complicated. Consequently, a specific segmentation techniques will be developed for the adaptation of the intonation characteristics of jingju singing.

Both frequency domain and time domain algorithms are employed in vibrato detection. In the former, we exploit the fact that the spectrum of the vibrato pitch contour represents a predominant peak in the considered frequency range [18]. In the latter, vibrato period is measured with the temporal distances between two local maxima or minima. The presence of vibrato is estimated by several parameters, such as mean and variance of vibrato period, number of distances [19]. [20] examined the vibrato statistics of two jingju singing role-types - *laosheng* and *dan*, which shows a similar vibrato rate and extent to Western opera singing.

3. APPROACH

Considering that the steady segment in jingju singing is much less 'stable' than that of Hindustani vocal music (Section 1), we develop here a new segmentation method (Figure 2). Firstly, we perform the preliminary segmentation by using a modified melodic transcription algorithm to obtain the pitch contour of each note. Secondly, we conduct the finer segmentation within each note's pitch contour by employing the standard deviation of the cumulative differences of local extrema (StdCdLe) as the criterion. Thirdly, we classify the segmented pitch contours into three categories: linear trend, vibrato and 'others'. Then, we perform a new round of segmentation only for the pitch contours in 'others' category, and reclassify the results into linear trend category. Finally, a straight line is fitted to the pitch contours of linear trend category by linear regression, and a refinement process is performed by concatenating the over-segmented pitch contours according to three criteria: the fitted slope, pitch and timing distances.

3.1 Preliminary Segmentation

We use Mauch's algorithm to extract the fundamental frequency contour (pitch contour) and segment it into musical notes [9]. With the aim of improving the melodic transcription accuracy specifically for jingju singing, we incorporate a genre-specific musicological model into the note tracking step: bigram note transition probabilities estimated from a jingju singing score dataset. Bigram note

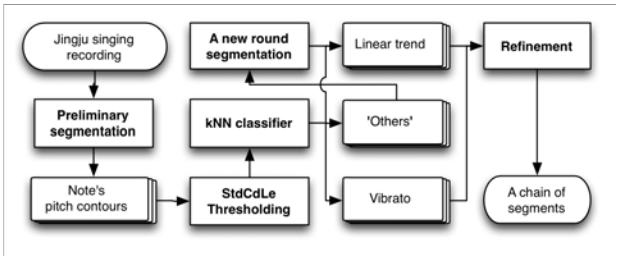


Figure 2. The block diagram of pitch contour segmentation.

transition probabilities define the probabilities of a transition from one note to another. Distinctive musical traditions represents distinctive bigram probabilities. For example, the major third note transition probability of jingju singing is much lower than that of Germany and Poland folk songs [11]. Unlike the probabilities estimated from a dataset, original algorithm provides an empirical, gaussian-shaped likelihoods which do not correspond to actual fact, for example, the excessively high probability of self-transition.

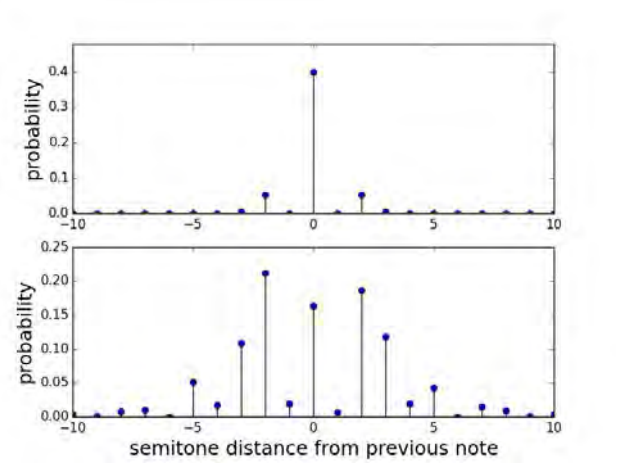


Figure 3. Central part of the note transition probabilities: original (upper) and those estimated from jingju singing score dataset (lower).

We estimate the bigram note transition probabilities (Figure 3) independently of the musical key from a jingju singing score dataset [1]. This dataset focuses on three jingju role-types: *dan* (female), *laosheng* (elderly man) and *jing* (painted-face). It contains 62 arias and represents 20827 notes. They are manually transcribed from printed sheet music into MusicXML format.

3.2 StdCdLe Thresholding

The strong variability of the jingju singing pitch contour (Section 1) urges us to reconsider the concept of each segment category. The "pure" steady segment as appeared in Hindustani vocal music is hard to find, which thus forces us to search for some uniformity among the variability during the segmentation process, such as the uniform rise and fall of a pitch contour. In this work, we use the criterion - StdCdLe to measure the uniformity of the variability of a segment, and assume that the StdCdLe of a segment should be lower than a given threshold (Figure 4).

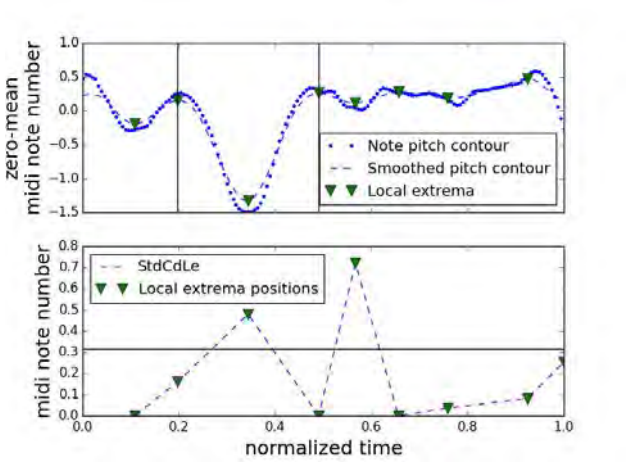


Figure 4. The note's pitch contour (upper): vertical black solid lines represent the segmentation points. The StdCdLe (bottom): horizontal black solid line represent the segmentation threshold.

The time interval of each contour obtained by the modified melodic transcription algorithm is normalized to [0,1]. Then, the pitch contour is subtracted by its mean value and passed through a moving average filter to remove the vocal jitter. After that, the local extrema in cents $A = \{\alpha_1, \alpha_2, ..., \alpha_M\}$ are detected on each pitch contour as the potential segmented points and its forward difference is denoted as $\triangle A = \{\alpha_2 - \alpha_1, ..., \alpha_M - \alpha_{M-1}\} = \{\delta_1, \delta_2, ..., \delta_{M-1}\}$ Finally, we segment the note's pitch contour on the basis of thresholding iteratively its StdCdLe. Whenever the StdCdLe exceeds a threshold th_s, the current local extremum is set as a segmentation point $B = \{\beta_1, \beta_2, ..., \beta_N\}$, and the cumulative vector C is reset to empty. Otherwise, the current difference is appended to the cumulative vector (Algorithm 1).

Algorithm 1 StdCdLe Thresholding Function

function StdCdLe($\triangle A$) : B
  $j \leftarrow 1, k \leftarrow 1$
  for $i \leftarrow 1, 2, ..., M-2$ **do**
    $C[j] \leftarrow \delta_i, j \leftarrow j+1$
    if std$(C) > th_s$ **then**
      $\beta_k \leftarrow i+1, k \leftarrow k+1$
      $C \leftarrow$ empty array, $j \leftarrow 1$
    end if
  end for
  return B
end function

3.3 Pitch Contour Classification

The aim of this step is to classify the preliminary segments into three categories: linear trend, vibrato and 'others'. The linear trend category is supposed to contain the segments with linear characteristic, such as steady and transi-

[1] https://github.com/jingju-SMC2016-PCS/SMC2016

tory ones. The 'others' category contains those which can be further segmented. The selected features are listed in Table 1.

Category	Features	Dim.
Linear regression	Regression coefficients	2
	R-squared	1
	Mean squared error	1
	Fitting curve crossing	1
Others	Vibrato rate	1
	Extrema number	1
	Contour length	1
	Standard deviation	1

Table 1. Pitch contour classification features and their dimensions (Dim.)

The linear regression feature category is used for describing the linearity of the segment, of which the 'fitting curve crossing' is calculated in a similar way as the 'zero crossing' but using the fitting curve as the crossing axis. This feature is designed for the classification of vibrato segment which tends to have a large 'fitting curve crossing' number. Due to the error of pYIN pitch estimation algorithm, the segment may contain outliers. We employ the measure of Cook's distance [21] to exclude the points in the segment which exhibit a large degree of influence on the estimated coefficients. The vibrato rate is calculated by using the frequency domain approach [18]. A kNN is chosen as the classifier considering its simplicity.

Due to the 'non linear' characteristic, the segments classified into 'others' category will be segmented a new round by simply using the local extrema as the segmentation points. The expected results will be reclassify into linear trend category which only consists of steady and transitory segments.

3.4 Refinement

The aim of the refinement step is to eliminate the phenomenon of over-segmentation which is mainly brought by the preliminary segmentation step (Section 3.1) in the edge region of the note's pitch contour. The approach is to concatenate the adjacent segments which meet certain given conditions (Figure 5).

We use the same linear regression technique mentioned in Section 3.3 to fit the segments, then extract their slopes. The slope, the starting and ending pitches of the fitted line of the segment i are respectively denoted as k_i, f_i^s and f_i^e. The starting and ending times of the segment i are respectively denoted as t_i^s and t_i^e. The concatenation conditions are listed as following:

$$|k_{i+1} - k_i| < th_{\text{slope}} \quad (1)$$

$$|t_{i+1}^s - t_i^e| < th_{\text{time}} \quad (2)$$

$$|f_{i+1}^s - f_i^e| < th_{\text{pitch}} \quad (3)$$

The condition 1 verifies that the adjacent segments both have the similar slope. The condition 2 and condition 3

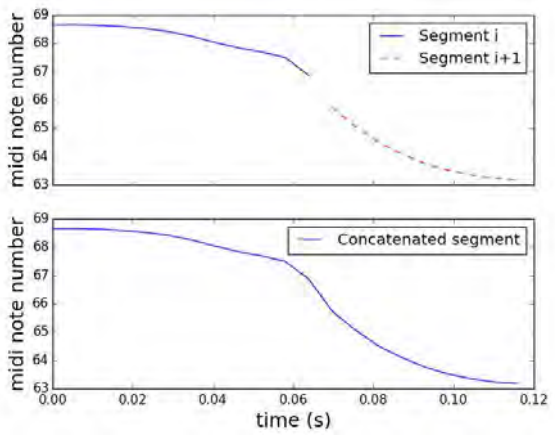


Figure 5. An example of concatenation of adjacent segments which meet the given conditions.

make sure that they are well connected in terms of boundary timing and pitch. If the adjacent segments are both flat pitch segments, that is to say:

$$|k_i|, |k_{i+1}| < th_{\text{flat pitch}} \quad (4)$$

only conditions 1 and 2 need to be fulfilled to achieve the concatenation. If the adjacent segments are both not flat pitch segments but have the same slope signs:

$$k_i k_{i+1} > 0 \quad (5)$$

all of the three conditions need to be fulfilled to perform the concatenation. If the adjacent segments are both not flat pitch segments and have the different slope sign, they can not be concatenated.

4. EVALUATION AND RESULTS

4.1 Dataset

The *a cappella* singing audio dataset used for melodic transcription and pitch contour segmentation tasks coming from MTG and C4DM [2] [22] focuses on two most important jingju role-types [23]: *dan* (female) and *laosheng* (elderly man). It contains 41 interpretations of 33 unique arias sung by 13 jingju singers. The melodic transcription, Std-CdLe thresholding, pitch contour classification and segmentation ground truth are manually annotated to all of the arias[1] . The melodic transcription ground truth represents 7686 notes. The average note duration is 0.38s and standard deviation of note duration is 0.37s. The pitch contour segmentation ground truth represents 14467 segments. The average segment duration is 0.21s and standard deviation of note duration is 0.25s. All of them are manually annotated in Sonic Visualizer [3] .

Since no parameter needs to be optimized in the preliminary segmentation step, all of the audio dataset and the annotated ground truth for the melodic transcription are used to evaluate its accuracy. For the rest of the steps, the dataset

[2] http://isophonics.net/SingingVoiceDataset
[3] http://www.sonicvisualiser.org/

Algorithms	COnPOff			COnP		
	F-measure	Precision	Recall	F-measure	Precision	Recall
Baseline	0.718	0.716	0.720	0.757	0.755	0.759
Modified	**0.730**	0.736	0.724	**0.769**	0.774	0.762

Table 2. Results for melodic transcription.

Algorithms	COnPOff			COnP		
	F-measure	Precision	Recall	F-measure	Precision	Recall
Baseline	0.284	0.307	0.264	0.534	0.592	0.487
Proposed	**0.388**	0.480	0.326	**0.642**	0.793	0.539

Table 3. Results for pitch contour segmentation.

is randomly split into 4 parts with the constraint that each part is selected without role-type bias and contains almost an equal number of segments. 3 of them are reserved as the training set for the purpose of parameter optimization. Another part is used as the test set to evaluate the pitch contour segmentation accuracy.

4.2 Evaluation Metrics

According to [9], note-based evaluation can expose more subtle details than frame-wise evaluation. We adopt the note-based evaluation metrics described in [24] for the melodic transcription evaluation: COnPOff and COnP. COnPOff takes into account correct note onset time (+/- 50 ms), pitch (+/- 0.5 semitones) and offset (+/- 20% of the ground truth note duration or +/- 50 ms, whichever is larger) is the most strict metric. COnP (correct note onset time and pitch) is the relax metric.

For the pitch contour segmentation evaluation, we adopt the similar metrics. However, since there is no demand for the pitch correctness, we simplify the metrics as: COnOff (correct segment onset and offset) and COn (correct segment onset).

4.3 Parameters Optimization

The parameters which need to be optimized are: StdCdLe threshold th_s in Section 3.2, the number of kNN neighbors K in Section 3.3 and th_{slope}, th_{time}, th_{pitch}, $th_{\text{flat pitch}}$ in Section 3.4. We use the grid search algorithm to perform the optimization. Since the kNN classification is a learning algorithm, a 3-fold cross-validation is adopted on the training set to measure its performance metric - misclassification rate. As K increases, the misclassification rate firstly goes down, then stabilizes. The optimal K is set at the beginning of the stable zone. The cross-validation technique is not employed during the optimization process of other parameters, because these steps do not contain learning algorithms and the performance metric - F-measure of COnOff can be report directly by sweeping these parameters on the training set without validation. Table 4 lists the search bounds or sets and the optimal results. The complete results of the grid search can be found on the web page[1] .

Parameters	Search bounds or sets	OR
th_s	$[0.01, 1.0]$ with step 0.01	0.22
th_{slope} (deg)	$[10, 90]$ with step 10	60
$th_{\text{flat pitch}}$ (deg)	$[10, 90]$ with step 10	30
th_{time} (sec)	{0.01,0.02,0.05,0.1,0.2,0.5}	0.02
th_{pitch} (semitone)	{0.1,0.2,0.5,1,2,5,10}	5
K	$[3, 20]$ with step 1	13

Table 4. Search bounds or sets, optimal results (OR) of the optimization process for each parameter.

4.4 Results and Discussion

For the melodic transcription algorithms evaluation, we run a modified version of Mauch's algorithm which incorporated with the jingju singing bigram note transition probabilities. The results (Table 2) show that the modified algorithm slightly outperforms the original one (baseline), which proves of the necessity of the incorporation of genre-specific musicological model into transcription system [12, 13].

For the pitch contour segmentation algorithms evaluation, we choose Ganguli's method [15] as the baseline. The proposed method largely outperforms the baseline method (Table 3) because, firstly, as we mentioned above, the steady segment in jingju singing contains more variability than that in Hindustani vocal music, thus the 'pitch histogram discretisation' approach used by Ganguli which only searches 'pure' steady segment fails; secondly, the baseline method is more likely to lose effectiveness on segmenting the vibrato in jingju singing which mainly rides on a steady segment, whereas that in Hindustani vocal music mainly appears as undulating glide - nearly periodic oscillation rides on a glide-like transition [17]. In general, the better adaptation of our method to jingju singing characteristics leads to a better segmentation accuracy.

5. WEB PROTOTYPE

The web prototype[4] (Figure 6) is designed for desktop browser and realized using mainly JavaScript. The data visualization and interaction are handled in the frontend, uti-

[4] https://dunya.compmusic.upf.edu/smc-2016

lizing primarily two JavaScript libraries D3.js and wavesurfer.js. The implementation uses Web Audio API and HTML5 technologies. The recorded audio from the user is sent to a backend server set up by Python, where the computation of the algorithms is done, and the results are visualized dynamically in the frontend.

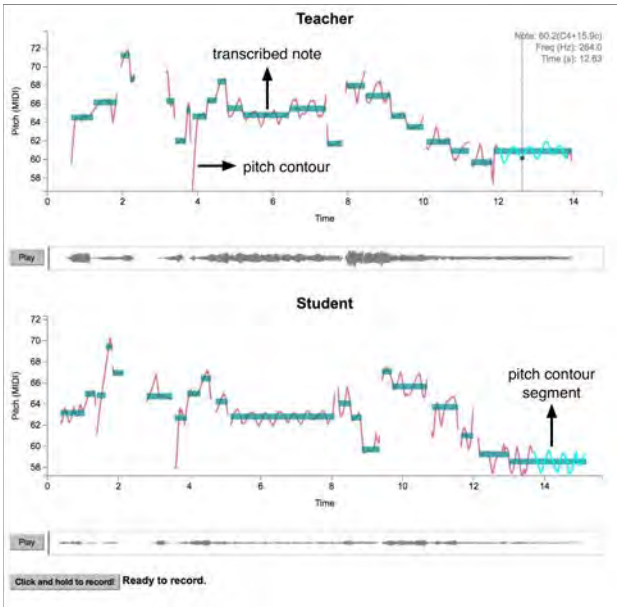


Figure 6. A web prototype for jingju singing training.

The web prototype consists of mainly two graphs, one of the teacher, which is the imitation target, and the other of the student, which reflects the students singing. By comparing the two graphs, the tool provides useful analysis to help the students find the difference and improve with practice. Each graph consists of a pitch contour of the singing melody, and notes derived from the pitch contour. Hovering over the mouse on the graph will highlight current pitch contour segment or note, and the aligned one on the other graph. The pitch (in Hz, cents and letter notation), duration and starting, end time information of the selected note and its aligned one will be shown on the up-right corner of each graph. Clicking and holding the record button will start recording. Releasing the record button the student graph will be updated with the analysis of the recording. A trainer and a student of jingju singing from NACTA (National Academy of Chinese Theatre Arts, China) have tried using this interface during the training class. They both indicated that it can provide a visual representation for the singing intonation and timing, and effectively help identify the problem in a rapid and intuitive way.

6. CONCLUSION AND FUTURE WORK

In this paper we have presented two techniques for computer-aided jingju singing training, their evaluations and a web prototype implementation.

The melodic transcription results suggest that the incorporation of the genre-specific musicological model - bigram note transition probabilities of jingju singing can successfully increase the transcription accuracy to this type of music in terms of note-based evaluation. The study of pitch contour segmentation shows that our proposed algorithm is able to search for the uniformity among the high variability included in the steady segments of jingju singing and achieve a better segmentation accuracy than the state of the art method. The web prototype provides intonation analysis with a fine granularity, and helps the students find the difference by comparison with the teacher.

With the deepening of our research on jingju singing, more musicological models or knowledge will be exploited to optimize the pitch contour segmentation algorithms. Since the syllabic level phonation accuracy occupies an important place in jingju singing training, the effort will be also focused on syllable-related techniques, such as syllable segmentation. Finally, the research of a better web interface which can effectively help in the training process and the systematic evaluation of its effectiveness will be included in our future works.

7. REFERENCES

[1] C. Mackerras, *The rise of the Peking Opera, 1770-1870: social aspects of the theatre in Manchu China*. Clarendon Press, 1972.

[2] E. Wichmann, *Listening to Theatre: The Aural Dimension of Beijing Opera*. University of Hawaii Press, 1991.

[3] E. Molina, "Automatic scoring of singing voice based on melodic similarity measures," Master thesis, 2012.

[4] W.-H. Tsai and H.-C. Lee, "An automated singing evaluation method for Karaoke systems." IEEE, May 2011, pp. 2428–2431.

[5] O. Mayor, J. Bonada, and A. Loscos, "The singing tutor: Expression categorization and segmentation of the singing voice," in *In Proceedings of the AES 121st Convention*, 2006.

[6] R. Schramm, H. d. S. Nunes, and C. R. Jung, "Automatic Solfège Assessment," in *International Society for Music Information Retrieval Conference*, Málaga, Oct. 2015.

[7] J. M. Geringer and C. K. Madsen, "Musicians' Ratings of Good versus Bad Vocal and String Performances," *Journal of Research in Music Education*, vol. 46, no. 4, pp. 522–534, Dec. 1998.

[8] E. Benetos, S. Dixon, D. Giannoulis, H. Kirchhoff, and A. Klapuri, "Automatic music transcription: challenges and future directions," *Journal of Intelligent Information Systems*, vol. 41, no. 3, pp. 407–434, 2013.

[9] M. Mauch, C. Cannam, R. Bittner, G. Fazekas, J. Salamon, J. Dai, J. Bello, and S. Dixon, "Computer-aided Melody Note Transcription Using the Tony Software: Accuracy and Efficiency," in *Proceedings of the First International Conference on Technologies for Music Notation and Representation*, May 2015.

[10] A. de Cheveigné and H. Kawahara, "YIN, a fundamental frequency estimator for speech and music," *The Journal of the Acoustical Society of America*, vol. 111, no. 4, pp. 1917–1930, 2002.

[11] T. Viitaniemi, A. Klapuri, and A. Eronen, "A probabilistic model for the transcription of single-voice melodies," in *2003 Finnish Signal Processing Symposium, FINSIG'03*, 2003, pp. 59–63.

[12] X. Serra, "A Multicultural Approach in Music Information Research," in *Proceedings of the 12th International Society for Music Information Retrieval Conference (ISMIR-2011)*, Miami (Florida), USA, Oct. 2011, pp. 151–156.

[13] A. Klapuri and M. Davy, Eds., *Signal Processing Methods for Music Transcription*. Boston, MA: Springer US, 2006.

[14] S. Gulati, J. Serrà, K. K. Ganguli, and X. Serra, "Landmark Detection in Hindustani Music Melodies," in *International Computer Music Conference/Sound and Music Computing Conference*, Athens, Greece, Sep. 2014, pp. 1062–1068.

[15] K. K. Ganguli and P. Rao, "Discrimination of melodic patterns in indian classical music," in *2015 Twenty First National Conference on Communications (NCC)*, Feb. 2015, pp. 1–6.

[16] S. S. Miryala, K. Bali, R. Bhagwan, and M. Choudhury, "Automatically Identifying Vocal Expressions for Music Transcription." in *International Society for Music Information Retrieval Conference*, 2013, pp. 239–244.

[17] C. Gupta and P. Rao, "Objective Assessment of Ornamentation in Indian Classical Singing," in *Speech, Sound and Music Processing: Embracing Research in India*. Springer Berlin Heidelberg, 2012, pp. 1–25.

[18] N. Kroher, "Automatic Characterization of Flamenco Singing by Analyzing Audio Recordings," Master thesis, Universitat Pompeu Fabra, 2013.

[19] S. Rossignol, P. Depalle, J. Soumagne, X. Rodet, and J.-L. Collette, "Vibrato: detection, estimation, extraction, modification," in *Digital Audio Effects Workshop (DAFx'99)*, 1999.

[20] L. Yang, M. Tian, and E. Chew, "Vibrato characteristics and frequency histogram envelopes in Beijing opera singing," *5th International Workshop on Folk Music Analysis*, pp. 139–140, 2015.

[21] R. D. Cook, "Detection of Influential Observation in Linear Regression," *Technometrics*, vol. 19, no. 1, pp. 15–18, 1977.

[22] D. A. A. Black, M. Li, and M. Tian, "Automatic Identification of Emotional Cues in Chinese Opera Singing," in *13th Int. Conf. on Music Perception and Cognition (ICMPC-2014)*, 2014, pp. 250–255.

[23] R. Caro Repetto and X. Serra, "Creating a Corpus of Jingju (Beijing Opera) Music and Possibilities for Melodic Analysis," in *15th International Society for Music Information Retrieval Conference (ISMIR-2014)*, Taipei, Taiwan, Oct. 2014, pp. 313–318.

[24] E. Molina, A. M. Barbancho, L. J. Tardón, and I. Barbancho, "Evaluation Framework for Automatic Singing Transcription," in *Proceedings of the 15th International Society for Music Information Retrieval Conference (ISMIR-2015)*, 2014, pp. 567–572.

ENGAGEMENT AND INTERACTION IN PARTICIPATORY SOUND ART

Visda Goudarzi
Institute of Electronic Music and Acoustics
University of Music and Performing Arts Graz, Austria
goudarzi@iem.at

Artemi-Maria Gioti
Institute of Electronic Music and Acoustics
University of Music and Performing Arts Graz, Austria
gioti@iem.at

ABSTRACT

This paper explores a variety of existing interactive and participatory sound systems and the role of different actors in them. In human computer interaction (HCI), the focal point on studying interactive systems has been the usability and functionality of the systems. We are trying to shift the focus more towards creative aspects of interaction in both technology development and sound creation. In participatory sound art, the roles of technology creator, composer, performer, and spectator are not always distinct but may overlap. We examine some challenges in such systems, like the ownership of technical and aesthetic components and balancing engagement and interaction among different stakeholders (designer, composer, spectator, etc). Finally, we propose a discussion on participation, human-computer and human-human interaction within the process of creation and interaction with the system.

1. INTERACTIVE SOUND SYSTEMS

The process of design and development of interactive sound systems used to be a separate task from sound creation. From design and development to performance, there used to be a linear flow (Figure 1). New interfaces for sound creation, however, allow sound artists and composers to engage themselves more in the process of system design and development.

In HCI and software development, an iterative approach of design and development is common where evaluations allow to improve the system in successive iterations. Applications of adapted iterative HCI methods in sound creation range from interaction design to creativity support in sound and technological domains. Studies which incorporate HCI methods to evaluate sound creation systems are mostly focused on how musical tasks are performed. Aspects evaluated might be related to performance [1], the quality of the user experience and the degree of expressiveness [2, 3, 4], the usefulness of the system [5] or participant's/audience's engagement [6]. Our approach aims to direct attention away from the system's accuracy and efficiency. We are more intrigued by non-quantifiable goals and notions such as creativity and engagement. In terms of interaction, we find Suchman's perspective [7] appropriate for interactive and particularly participatory sound systems. She moves away from a goal-oriented interactivity and meaningful action towards a concept of interactivity in which action is central and goals are emergent. Furthermore, the iterative development of software engineering is not necessarily transferrable to artistic creation. In artistic production iterative processes are part of the creative experimentation and not part of the evaluation of a completed artwork [8].

1.1 Stakeholders and the scope of interaction

In traditional sound making systems (e.g. musical instruments), the designer of the sound system was usually a different person than the musicians (composers and performers). Due to rapid technological advancements, the gap between designers and sound makers is getting smaller. Examples include the democratization of sound and musical instruments, the evolution of Internet, open source software, community based design and DIY (Do It Yourself) instruments. However, sound making is usually left to composers and musically trained people. There are only a few sound artworks that allow people with little or no experience in sound or music to participate in the creative process and potentially increase their awareness about sound in their surroundings improving their analytical listening skills [9].

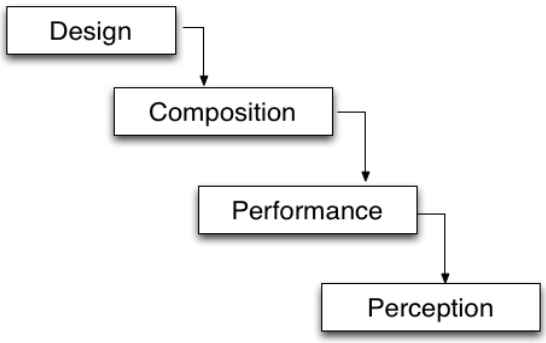


Figure 1. An overview of the flow of actions in a traditional interactive sound system.

Furthermore, in the last decade computing systems such as tablets and mobile devices, which used to be specifi-

cally deployed by engineers and experts, have become more popular among the general public. This has led to a different media ecology, extending the cultural context for interactions through consumer devices and creating a new platform for engagement in participatory art, by using one's own devices.

Having such powerful -potentially sonic- devices in hand, the space becomes the instrument and the stage of creativity that has so far been confined to a small elite of educated musicians expands [10]. In the last decade, research has focused on the development of methods to assist music composition by hiding part of the complexity of creation from the user [11]. This is on the one hand a democratization of sound or any content creation, but on the other hand it can add to the sonic pollution in public spaces. You may hear up to 20 different sounds within a minute coming out of a mobile phone because of different apps. At the same time this has created a new condition for artistic creation: art is not presented only in museums and galleries; it has rather expanded to public spaces much easier and faster than before.

1.2 Design and Development

HCI design process could be a goal-oriented problem solving activity informed by intended use, a creative activity, or a decision-making activity to balance trade-offs (e.g. requirements of product compatibility and ease of use may be contradicting.) In a HCI design process there is usually a plan for development and a set of alternatives and successive elaborations. Whether the main design decisions are made by the composer or the designer depends on the goals and complexity of the system. Available computer languages could also have a huge influence on the focus of the design process. McCartney [12] describes the early computer music languages as strong in abstractions but providing very few control and data structures and little or no user functions. Later on, computer music languages such as Max and Pd allowed abstractions, so that in some cases the users are not even noticing that they are programming. These languages allowed for more participation of composers and performers in the process of programming sound systems, without requiring extensive software engineering training. Most importantly, the orientation of these programming languages towards live interaction brought HCI into the focus of both composition and performance, bringing about important changes in both fields.

1.3 Composition and Performance

The growing availability of electronic devices and audio programming languages has lead not only to a merging of instrument/interface design and composition, but also to a new understanding of composition and performance as interactive processes. Human-computer interaction in the composition and performance of live electronic music ranges from interface-based interaction, employing human-computer interfaces that translate the performer's actions into sound, to responsive and autonomous agent-based systems, able to interact with human performers in real-time.

The latter approach in particular has broadened our understanding of composition and the traditional roles of composer and performer. By delegating some responsibility to machine agency, the real-time interaction between the performer and the computer is transformed into a dynamical and reciprocal process of communication between human and machine agency. Due to this integration of human and machine agency the spectrum between "fixed" composition and free improvisation is becoming "increasingly densely populated" [14], while the borders between performer and composer, as well as instrument and composition are becoming obscure.

An example of the merging of the traditional roles of composer and performer is interactive composing. Chadabe describes interactive composing as 'a two-stage process that consists of (1) creating an interactive composing system and (2) simultaneously composing and performing by interacting with that system as it functions' [15]. In interactive composing systems, also known as real-time composition systems [16], the composer is also the performer, while it is impossible to distinguish between composition and performance, since they occur simultaneously.

A different approach to interaction in the composition of live electronic music are Di Scipio's audible ecosystems, in which the field of interaction is defined as the triangular connection among a human agent, a DSP unit and the sonic ambience [17]. In Di Scipio's *audible ecosystemics* interaction is expanded to include not only the human agent and the computer, but also the ambience itself as a performance agent.

Further examples of interaction strategies in live electronic music performance are performance networks and virtual player systems. Performance networks are technology-based systems that enable remote or co-located collaborative musical performance. Weinberg [18] categorises interconnected musical networks in three different approaches: the *server approach*, in which the network is limited to the individual interaction between each player and the system, the *bridge approach*, that enables collaborative performances among performers that are in different locations, and the *shaper approach*, in which the system uses algorithmic processes in order to generate musical material that the performers can modify collaboratively.

Virtual player systems are systems that fall into the *player* category of Rowe's taxonomy [19]. In these systems, a virtual player learns from one or more human improvisers and responds to them in real-time. Examples of virtual player systems are those developed by Lewis [20], Bakht and Barlow [21] and Dubnov et. al [22]. Such systems usually employ machine learning in order to produce musical material that is similar to that played by the human improviser. For this reason, the material generated

by the virtual player is in most cases pitch-based. This, in addition to their machine learning based approach, makes such systems more suitable for improvised performances, than for compositional applications.

Finally, a machine learning, but not pitch-based approach was followed by the creators of the *Wekinator* software, designed for end-user interactive machine learning [23]. With the Wekinator, the user (performer/composer) can create desired gesture-sound mappings by training a learning algorithm. The software is end-user oriented, enabling musicians to work with intelligent systems without requiring any programming knowledge. However, its approach is in fact interface-based: even though it uses machine learning, its functionality is restricted to gesture-sound mappings.

1.4 Audience participation and engagement

Expanding the field of interaction beyond composition and performance to audience participation has usually overlapped with the creation of multi-user instruments, which have switched the role of a passive audience (or spectator) to an active player. Dixon [24] identifies four types of interaction based on different levels of engagement: navigation, participation, conversation and collaboration.

Engaging the audience is not a new approach in sound art, but engaging them to the extent of being the main creators has not been explored in depth. E.g. John Klima's *Glasbead* [25] is a great example of multi-user collaborative musical interface used to engage 20 or more remote players with each other. Another example is Golan Levin's *Telesymphony* [26], in which he choreographed ringing of audiences' mobile phones. In this example, the audience doesn't have any control over the structure of the piece or the creation of the sounds. They are almost passive users with an active instrument in hand that is mostly controlled by the composer and creator of the piece.

In recent years, since crowd sourcing and creating content by users have become more common, interfaces that take advantage of that also entered the music world, such as Kruge's *MadPad* [27]. He uses the audio/visual content that the audience sends before the performance, during the concert. In this approach, the users create the whole content and the performer only uses algorithms to compose with it. However, the audience is not participating in real time. Other real-time applications are: *Tweetscapes* [28] and *TweetDreams* [29]. *Tweetscapes* is a project of sonification experts, media artists, and a radio broadcaster. Online data from German Twitter streams is sonified and visualized in real-time. The sounds are based on a large sound database and randomly – but reproducibly fixed – assigned to different semantic terms (hashtags). These sounds are then modified according to metadata, e.g. from which location in Germany the tweet was sent. Another example is *TweetDreams* by Dahl et.al. which is a sonification of tweets from the audience played by a laptop ensemble. The piece is not as precisely choreographed as *Telesymphony*, which gives the audience a certain freedom to "compose" (at least composing their own tweets) within the framework of the piece. Still conducting the piece is left in the hands of the designer. The individual cannot change the direction of the whole structure of the ensemble, but has at least control over his/her own sounds (or tweets). Ximena Alarcon's [30] *Sounding Underground* is an example of leaving the creative aspects to the user. It's an online interactive sonic environment which links sound experts from metros of London, Paris, and Mexico City. She translates the public transport into interactive multimedia using interactive ethnography to involve participants' perception of space.

2. ASPECTS OF AUDIENCE PARTICIPATION

From these examples it is clear that participatory sound systems display a wide range of both participation and interaction strategies. Some of the most important parameters of participatory sound systems design are discussed in detail below:

2.1 Audience engagement

Harries [31] refers to authorship, performance and spectatorship as different types of audience participation using the terms performance and authorship as interchangeable. We would like to distinguish between three types of audience participation, with increasing level of engagement: crowdsourcing, performance agency and co-authorship.

Crowdsourcing, in general, is a paradigm for the use of human processing power to solve problems. Computational systems where the crowd performs tasks to solve problems in the context of computer music are very common, especially in the field of music information retrieval. To name a few: *Mechanical Turk* that uses people's opinion to find similarities between songs [32], *Last.fm* [33] and *Freesound* [34] that use the crowd sound sample collection and music library management. In participatory systems crowdsourcing refers to audience-computer interaction systems that allow a large crowd to participate in the process of sound making mainly by functioning as a source of data, such as *TweetDreams* [29], *Flock* [35] and *One Man Band* [36]. For instance, in *TweetDreams*, audience members use their personal mobile devices to tweet. Tweets containing hash-tags (chosen by performers) are sonified and visualized into a dynamic network.

In *performance agency* the role of the audience is similar to that of a performer. Unlike crowdsourcing, in performance agency audience engagement is active and involves real-time control over sound parameters that are determined by the composer/designer. Even though com-

positional decisions are made by the creator of the system, the audience is able to explore the 'space' defined by the composer/designer and interact with it by setting runtime control data. However, despite the active participation, performance agency as a form of audience engagement implies a hierarchically structured interaction model, based on the dichotomy between the creator/designer and spectator/performer. An example of performance agency as an audience engagement strategy is *Auracle*. In *Auracle* the users can control a synthesized instrument with their voice, while interacting with other users in real time over the Internet [37].

The most active form of audience participation is *co-authorship*. In this case, the spectator is not just a performer, but co-author in the process of sound creation. Instead of setting the values of a fixed set of run-time control variables, the audience is invited to participate in the creative process, by making compositional decisions regarding both the sound material and the processes applied to it. In this case, the designer's role is limited to the creation of a platform that enables collaborative sound creation, while it involves little to no compositional responsibility at all. Co-authorship is the most democratic form of audience participation and the one that has been explored the least by participatory systems so far.

2.2 Human-computer interaction

Our discussion of human-computer interaction in participatory sound systems focuses on the aspects of control and mapping. Instead of discussing technical aspects of HCI, like functionality or usability, we prefer to focus on different strategies in the design of audience-system interaction in participatory systems. The aspects of HCI that we examine here are: 'multiplicity of control' [14], type of control, mapping of control actions, control parameters and 'control modality' [14].

- ***Multiplicity of control***: By multiplicity of control we refer mainly to the differentiation between single-user and multi-user systems. In single-user systems, only one user can interact with the system at a given moment, while in multi-user systems more than one users can interact with the system simultaneously. The concept of multiplicity of control is not necessarily limited to human agency, but can also include machine agency, meaning that the system itself can perform control actions. An example of an interactive sound system with multiple control channels is the *reacTable* [27].

- ***Type of control:*** The type of control refers to the different human-computer interfaces that can be used as part of the audience-system interaction. Like in live electronic music performance, this interaction can be tangible and embodied (e.g. Michael Waiswisz, The Hands) or disembodied (e.g. Alvin Lucier, Music for solo performer), non-tactile etc.

- ***Mapping of control actions:*** Mapping control actions to sound parameters is perhaps the most important part of sonic human-computer interaction design. Mapping processes can be linear (a simple scaling of input values to control values) or non-linear and based on dynamical processes (e.g. dynamical systems modeling, machine learning etc.). An example of a linear mapping process is assigning the keys of a MIDI keyboard to pitches, while an example of a dynamical mapping process, based on machine learning, is the software *Wekinator* [23]. The type of the mapping process affects the level of perceived control and transparency (i.e. how perceivable the relationship between control actions and sound output is).

- ***Control parameters***: Sound parameters, the value of which can be set by the user.

- ***Control modality***: Control modality refers to the type of control value (discrete or continuous) [14] and depends on the control parameter itself, as well as the type of the interface. For example, faders allow for continuous control and are more suitable for controlling a parameter like amplitude, while buttons could be used for triggering pre-recorded samples.

2.3 Human-human interaction

Human-human interaction in participatory sound systems has evolved excessively with the growth of internet and social media. Server-based cloud computing has enabled the audience to participate in performances in active or passive roles without the need for special technical background or pre-configuring hardware/software. The cost effectiveness, familiarity, and ease of use of some technological devices have also made the entrance to participation easier. Furthermore, the roles of performer, composer and audience have become more interchangeable and the participation has added more uncertainty factor to performances which is on the one hand technologically and artistically challenging, and on the other hand compelling. Some factors that influence the communication and interaction between audience members, or the audience and other performers are:

- ***Location:*** according to Barbosa [38] collaboration in participatory performances could take place within remote or co-located network music systems. In the former, people could participate in the performance from different parts of the world (e.g. *SoundWIRE* group [39]) or could be even in the same room using different computers connected to local networks. In both cases they share the same sonic environment. Since these systems create a platform of synchronous improvisations for a broad audience, participants usually don't need to be experts. Another possibility is that participants even share the same physical interface or device. (e.g. in table-top instruments such as the *reacTable.* [27])

- ***Levels of communication:*** some collaborative environments only allow participation of expert musicians in the participatory performance whereas some others encour-

age collaboration between experts and novices. In the latter, the communication is more focused on a performative involvement of the audience. In other cases a master performer reacts or communicates back to the audience [40] during the performance by shaping sounds or data received from the audience (hierarchical collaboration). Another level of communication is the interaction among audience members. However, what determines a successful sonic Human-Computer Interaction and how can participatory design encourage audience engagement are questions that still need to be answered. Especially, in the case of co-authorship participatory design seems to be compelling, both aesthetically and technologically.

3. DISCUSSION

In this paper we discussed some challenges in participatory and interactive sound systems, focusing on audience participation and engagement. Technological advancements and new artistic concepts have lead to a closer collaboration among the traditionally distinct fields of design, composition and performance and enabled various forms of audience participation. Participatory strategies in sound art can vary from a passive participation in which the audience functions simply as a source of data (*crowdsourcing*) to active participation in the performance and creation of an artwork (*performance agency* and *co-authorship*). The last two approaches are as radical as they are challenging, bearing implications for both the spectator and the creator/author. Questions regarding authorship inevitably arise as a result of this shift of creative responsibility: is a participatory sound work the work of the artist who designed it or is it a creation of the participants? How does this democratization of the creative process affect its goal? Is the goal of the creative process still the artifact or does the goal shift from the aesthetic artifact to the interaction itself? And, finally, how does collective creative responsibility affect the aesthetics and perception of the artwork?

4. REFERENCES

[1] M. M. Wanderley and N. Orio. “Evaluation of Input Devices for Musical Expression: Borrowing Tools from HCI,” *Computer Music Journal*, 26(3), pp. 62-76, 2002.

[2] D. Stowell, M.D. Plumbley and N. Bryan-Kinns, “Discourse analysis evaluation method for expressive musical interfaces,” in *Proceedings of the International Conference on New Interfaces for Musical Expression*, NIME 2008.

[3] C. Kiefer, N. Collins and G. Fitzpatrick, “HCI Methodology For Evaluating Musical Controllers: A Case Study,” in *Proceedings of the International Conference on New Interfaces for Musical Expression*, NIME 2008.

[4] O. Bau, A. Tanaka and W.E. Mackay, “The A20: Musical Metaphors for Interface Design,” in *Proceedings of the International Conference on New Interfaces for Musical Expression*, NIME 2008.

[5] T. Coughlan and P. Johnson, “Interaction in Creative Tasks: Ideation, Representation and Evaluation in Composition,” in *Proceedings of the ACM Human Factors in Computing Systems*, CHI 2006.

[6] N. Bryan-Kinns and F. Hamilton, “Identifying mutual engagement,” *Behaviour & Information Technology*, 31(2), pp. 101-125, 2009.

[7] L.A. Suchman, *Plans and Situated Actions: The Problem of Human-machine Communication*. Cambridge: Cambridge University Press, 1987.

[8] Z. Bilda, E. Edmonds and L. Candy, “Designing for creative engagement,” *Design Studies*, 29(6), pp. 525-540, 2008.

[9] R.I. Godøy, “Gestural-Sonorous Objects: embodied extensions of Schaeffer's conceptual apparatus,” *Organised Sound,* 11(02), pp.149-157, 2006.

[10] D. Cope, *Computer models of musical creativity*. Cambridge: MIT Press, 2005.

[11] H.C. Jetter, F. Geyer, T. Schwarz, and H. Reiterer “Blended Interaction-Toward a Framework for the Design of Interactive Spaces,” in *Workshop DCIS*. Vol. 12, 2012.

[12] J. McCartney, “Rethinking the computer music language: SuperCollider,” *Computer Music Journal*, 26(4), pp. 61-68, 2002.

[13] M. Wooldridge and N. Jennings, “Agent Theories, Architectures, and Languages: a Survey,” in *Intelligent Agents,* M. Wooldridge and N. Jennings, Ed. Berlin: Springer-Verlag, pp. 1-22, 1995.

[14] J. Pressing, “Cybernetic Issues in Interactive Performance Systems,” in *Computer Music Journal*, 14 (1), pp. 12-25, 1990.

[15] J. Chadabe, “Interactive Composing: An Overview,” in *Computer Music Journal*, 8 (1), pp. 22-27, 1984.

[16] A. Eigenfeldt, “Real-time Composition as Performance Ecosystem,” in *Organised Sound*, 16 (2), pp. 145-153, 2011.

[17] A. Di Scipio, “Sound is the interface: From interactive to ecosystemic signal processing,” in *Organised Sound*, 8 (3), pp. 269-277, 2003.

[18] G. Weinberg, “Interconnected Musical Networks: Toward a Theoretical Framework,” in *Computer Music Journal*, 29 (2), pp 23-39, 2005.

[19] R. Rowe, *Interactive Music Systems: Machine Listening and Composing*. London: MIT Press. 1993.

[20] G.E. Lewis, “Too Many Notes: Computers, Complexity and Culture in "Voyager",” *Leonardo Music Journal*, 10, pp. 33-39, 2000.

[21] S. Bakht, and C. Barlow, “PAPAGEI: An Extensible Automatic Accompaniment System for Live Instrumental Improvisation,” in *Proceedings of the International Computer Music Conference*, ICMC 2009 pp. 521-523, 2009.

[22] S. Dubnov, G. Assayag, O. Lartillot and G. Bejerano, “Using Machine-Learning Methods for Musical Style Modeling,” *IEEE Computer*, 10 (38), 2003.

[23] R. Fiebrink and D. Trueman, “End-User Machine Learning in Music Composition and Performance,” Present. CHI 2012 Work. End-User Interact. with Intell. Auton. Syst. Austin, Texas, May 6, 2012, pp. 14–17, 2012.

[24] S. Dixon, *Digital Performance: A History of New Media in Theater, Dance, Performance Art, and Installation*. Cambridge, MA and London: MIT Press, 2007.

[25] C. Paul, “Renderings of digital art,” *Leonardo*, 35(5), pp. 471-484, 2002.

[26] G. Levin, “A personal chronology of audiovisual systems research,” in *Proceedings of the International Conference on New Interfaces for Musical Expression*, NIME 2005, pp. 2-3, 2005.

[27] S. Jordà, “Multi-user instruments: models, examples and promises,” in *Proceedings of the International Conference on New Interfaces for Musical Expression*, NIME 2005, pp. 23-26, 2005.

[28] T. Hermann, A. V. Nehls, F. Eitel, T. Barri, and M. Gammel, “Tweetscapes - real-time sonification of twitter data streams for radio broadcasting,” in *Proceedings of the International Conference on Auditory Display,* ICAD 2012.

[29] L. Dahl, J. Herrera, and C. Wilkerson, “Tweetdreams: Making music with the audience and the world using real-time twitter data,” in *Proceedings of the International Conference on New Interfaces for Musical Expression*, NIME 2011, pp. 272-275 2011.

[30] X. Alarcón Díaz, “An interactive sonic environment derived from commuters' memories of the soundscape: a case study of the London Underground,” Ph.D. dissertation, Music, Technology and Innovation Research Center, De Montfort University, Leicester, United Kingdom, 2007.

[31] J. H. Lee, “Crowdsourcing Music Similarity Judgments using Mechanical Turk,” in *Proceedings of the International Society for Music Information Retrieval Conference,* ISMIR 2010, pp. 183-188, 2010.

[32] G. Harries, “‘The open work’: ecologies of participation,” *Organised Sound*, 18(01), pp. 3-13, 2013.

[33] X. Hu, M. Bay and J. S. Downie, “Creating a Simplified Music Mood Classification Ground-Truth Set,” in *Proceedings of the International Society for Music Information Retrieval Conference,* ISMIR 2007, pp. 309-310, 2007.

[34] F. Font, G. Roma, and X. Serra, “Freesound technical demo,” in *Proceedings of the 21st ACM International Conference on Multimedia*, pp. 411-412, 2013.

[35] J. Freeman, and M. Godfrey, “Creative collaboration between audiences and musicians in Flock,” *Digital Creativity*, 21(2), pp. 85-99, 2010.

[36] J. N. Bott, J. G. Crowley and J. J. LaViola Jr., “One Man Band: A 3D Gestural Interface for Collaborative Music Creation,” in *Proccedings of the Virtual Reality Conference*, VR 2009, pp. 273-274, 2009.

[37] J. Freeman, K. Varnik, C. Ramakrishnan, M. Neuhaus, P. Burk and D. Birchfield, “Auracle: a voice-controlled, networked sound instrument,” *Organised Sound*, 10(03), pp. 221-231, 2005.

[38] Á. Barbosa, “Displaced soundscapes: A survey of network systems for music and sonic art creation,” *Leonardo Music Journal*, 13, pp. 53-59, 2003.

[39] J. P. Cáceres and C. Chafe, “JackTrip: Under the hood of an engine for network audio*,” Journal of New Music Research*, 39(3), pp. 183-187, 2010.

[40] G. Paine, “Interactivity, where to from here?,” *Organised Sound*, 7(03), pp. 295-304, 2002.

GESTURAL CONTROL OF WAVEFIELD SYNTHESIS

Francesco Grani[1], Diego Di Carlo[2], Jorge Madrid Portillo[3], Matteo Girardi[4], Razvan Paisa[5], Jian Stian Banas[6], Iakovos Vogiatzoglou[7], Dan Overholt[8], Stefania Serafin[9]
SMC, Aalborg University Copenhagen
{[1]fg, [8]dano, [9]sts}@create.aau.dk
{[2]ddicar16, [3]jmadri15, [4]mgirar15, [5]rpaisa11, [6]jbanas14, [7]ivogia00}@student.aau.dk

ABSTRACT

We present a report covering our preliminary research on the control of spatial sound sources in wavefield synthesis through gesture based interfaces.

After a short general introduction on spatial sound and few basic concepts on wavefield synthesis, we presents a graphical application called *spAAce* which let users to control real-time movements of sound sources by drawing trajectories on a screen. The first prototype of this application has been developed bound to WFSCollider, an open-source software based on Supercollider which let users control wavefield synthesis. The *spAAce* application has been implemented using Processing, a programming language for sketches and prototypes within the context of visual arts, and communicates with WFSCollider through the Open Sound Control protocol. This application aims to create a new way of interaction for live performance of spatial composition and live electronics.

In a subsequent section we present an auditory game in which players can walk freely inside a virtual acoustic environment (a room in a commercial ship) while being exposed to the presence of several "enemies", which the player needs to localise and eliminate by using a Nintendo WiiMote game controller to "throw" sounding objects towards them. Aim of this project was to create a gestural interface for a game based on auditory cues only, and to investigate how convolution reverberation can affects people's perception of distance in a wavefield synthesis setup environment.

1. INTRODUCTION

The evolution of audio technology allowed for new listening setups to be experimented and evaluated. Long gone are the days of Thomas Edison's phonograph in 1877. Without doubt a milestone in the history of audio engineering, Edison's invention was able to both record and playback sound, however spatial fidelity was rather underwhelming, as the entire process was monophonic. Notably, not long after phonograph introduction, in 1881, a stereophonic playback device called the théâtrophone has been proposed by Clement Ader. The principle was simple - two microphones were placed across the opera stage and the signal collected by them was output to a pair of telephone receivers, placed in the opera house's foyer [1]. Later extensive research in this field slowly lead towards the commercial use of stereophony [2]. For some purposes it has been enhanced with an addition of a central speaker - mainly in cinemas, due to large dimensions of the screens. In consumer grade applications, stereophony has started to become widespread in the late 1950s with the invention of methods to engrave two channels onto a vinyl disc. At the same time, spatialisation of sound sources is an expressive tool that music composers had put into use since centuries. Dozens are the compositions of the 16th century Italian composer Giovanni Luigi da Palestrina that make use of spatial distribution of musicians. With the rise of the era of electronic music during the second half of the 20th century, the number of composers who pushed the boundaries of the available techniques in order to pursue their creative needs in terms of spatial sound just increased, often leaving commercial solutions behind the "brute force" *ad-hoc* methods adopted by composers and their sound technicians (just to mention few cases: Karlheinz Stockhausen's *Gesang der J´unglinge* (1955), Varese's *Poeme Electronique* (1958)). In particular cases artistic needs ended up in the construction of dedicated venues such as the Acousmonium, designed in 1974 by Francois Bayle to host spatialised sound concerts [3]. In most cases however the bridge between science and art has been very short, leading to various experiments in the field of recording and mixing techniques (e.g., [4]) that quickly brought us to a series of available techniques for the recording and reproduction of almost any desided sound field. Results achieved in sound spatialization techniques for systems of loudpseakers span today from stereo panning to more extended multichannel configurations, such as ITU 5.1 Surround [5], VBAP [6] and [7], DBAP [8], ViMiC [9], first and higer orders of Ambisonics [10], [11], as well as this project focus, wavefield synthesis (WFS) [12], [13].

Of all the above mentioned techniques, however, only WFS let in principle the listener to perceive the same designed soundfield in the same way and with the same auditory perspective from any point in space. This peculiar characteristic makes WFS a privileged technique to be adopted in situations like the ones described in this work.

2. BACKGROUND

The use of wavefield synthesis to recreate 3D sounds is not something new, having been pioneered in the late 80s at

the TU Delft University, Netherlands [12]. In recent years, with the increase in availability of computational power the technology has gained a commercial interest, supported by hardware and software solutions such as the ones proposed by IOSONO [14] and Sonic Emotion [15], [16] as well as several open source engines available to control, simulate and render WFS [17], [18], [19], [20].

2.1 Basics of Wavefield Synthesis

The foundation of WFS lies on the theory concept of Christiaan Huygens: *Each point on a wave front can be regarded as the origin of a point source. The superposition of all the secondary sources form a waveform which is physically indistinguishable from the shape of the original wave front* [21].

The principle has been originally used to describe water and optical waves, and was first formulated for acoustics in 1988 at the TU Delft after being pioneering described in the 50s by Snow et al. [2]. A WFS system does require a large number of loudspeakers, placed as close as possible to the next one in order to create an array with as few discontinuity as possible. Each loudspeaker of the array corresponds this way to a secondary sound source and needs to be driven by a dedicated/independent signal thus requiring a large number of audio channels, equal to the number of loudspeakers; the signal for each channel is calculated by means of algorithms based on the Kirchhoff-Helmholtz integrals and Rayleigh's representation theorems [22], [23]. Due to the physical and software limitation, WFS systems are enduring several approximations, which introduce certain limitations and artifacts.

A first approximation needed to minimize complexity is to reduce the control of the sound field from a 3D to a 2D space (an horizontal -unlimited- plane). A second approximation consists into limiting the amount of secondary sources to a finite number (a finite set of loudspeakers); this approximation leads towards the consequence that the frequency range whereas a WFS system provides artifacts-free sounds gets reduced to the portion of the acoustic spectrum that is located below a threshold frequency, named "spatial aliasing frequency"; above this frequency artifacts will occur in the form of "ghost sound images". To cope with this limitation it is desirable to place the loudspeakers at the minimum possible distance -in our case 16.4cm, thus introducing a spatial aliasing threshold of 1048 Hz- and to design a sonic content which is not unbalanced towards hi frequencies. Another approximation consists on the fact that linear arrays of loudspeakers have a limited physical length and this generates what is called "truncation error", a phenomenon that limits the angles of incidence of sound sources in which a good result of WFS can be achieved. Further interferences can be introduced by the loudspeaker construction itself, as well as by the acoustics of the room in which the system is installed. An exhaustive description of WFS limits can be found in [24].

In wavefield synthesis technique it is usual to distinguish between three categories of sound sources that can be reproduced.

- Point sources: virtual sources that are placed anywhere outside the inner area of the loudspeakers array.
- Plane waves: sources that are ideally placed at an infinite distance, thus their incident wavefront can be described as plane.
- Focused sources: sound sources that are located inside the area covered by of the loudspeakers array.

3. SPAACE

3.1 Introduction

The spatial characteristic of a composition has been an important topic for the avant-garde musicians in the past decades [25] and it still is a relevant quality of a musical piece [18]. From an artistic point of view, conveying spatial musical ideas and thoughts could underestimate the technical issues that must be faced during the development and implementation process of a software for musical purposes. Hence, contemporary composers and sound engineers have to find a trade-off during such process of composing new musical material. Moreover, learning new technologies or softwares for spatial music could be time consuming for composers that do not have a deep knowledge in the computer music field. Therefore, the *spAAce* application attempts to provide the following advantages:

1. a quick way to sketch and test movements of sound sources.
2. improvisation with spatial sound sources during live performances.

The latter point could lead to new approaches of performing live concerts in a live electronics scenario. Indeed, WFSCollider is employed just as engine render while the composers can focus on creating trajectories for sound sources and expanding musical expression.

3.2 State of the art

As introduced in 2, several different softwares for spatial sound movements have been developed in the recent years [26] [27] [28] and several spatial techniques have been implemented such as VBAP, DBAP, Ambisonics and Wavefield synthesis (WFS). Most of the spatial rendering engines come with a Graphical User Interface, such as SoundScape Render [28], Spat [26], WFSCollider [27] itself and others.

With the spreading of user-friendly GUI development environments for mobile and web app, some applications have been developed for these rendering engines, which allow real-time finger-based interaction. Some of them are more are mixing-oriented, providing a real-time positioning of the sound sources in the space, while others, such as Trajectoires [29] and Spatium [30], allow the users to move the sound sources and create complex paths both in time and in space. However, these new applications are still in the embryonic stage and there is a lot of work still to be done in order to design the most suitable interface that can capture the actual intentions of the performers. The key

concept of *spAAce* is the combination of several modes of interaction that should encourage and enable artistic sound spatialization.

3.3 Design process

Knowing what kind of technologies are usually employed, the development path has led us to employ Processing [31] as development platform, since it has many libraries and a strong community of developers. The next step has been asking to composers and sound engineers for interviews to test the concept of the *spAAce* application and to receive general feedback, hints and suggestions. The composers and sound engineers reached are highly involved in contemporary music production and live electronics performances. These interviews gave a solid starting point to implement the main core of the application. An iterative procedure of *implementation - testing - bugfixing* has been then employed for the development of the application.

3.4 Software architecture

The graphical user interface has been developed to be controlled by a multi-touch screen like a tablet or similar. The interface allows the user to create, control and delete trajectories for the sound sources. There are three types of trajectories:

- line trajectory
- circular trajectory
- free hand drawing

These trajectories are displayed as buttons on the left upper corner of the screen. Sound sources can be dragged with a finger and when placed on the top of one trajectory, getting automatically to follow the trajectory's path with a default speed. This speed can be changed with a knob on the bottom right side of the screen. Additionally, there is a control panel on the bottom left side of the screen where users can select each sound source to create control groups and perform mass editing. Lastly, since the OSC protocol is employed in order to communicate with WFSCollider, the users can set the proper IP Address and OSC port by clicking on the "Network" button located on the right upper side of the screen.

3.5 Device and Controller

The controller is not constrained to a particular hardware since the development has been done in Processing, which is a multi-platform application. In our testing sessions, a Wacom 22" multi-touch screen has been used, and a Leap Motion has been added in order to track hand movements and perform specific actions, such as moving sound sources towards a particular direction, applying spatial effects or even to control more than one sound source at the same time. Overall, the controller is designed so that the expression in terms of spatialization can be improved as much as possible.

Figure 1. The *spAAce* architecture.

3.6 Physical Setup

Since *spAAce* is a brand new project, it is continuously tested in the Multi-Sensory Experience Laboratory of Aalborg University in Copenhagen. All the experiments have been conducted with the following setup: A WFS system of 64 loudspeakers - presented with more details in 4.3; a computer running WFSCollider server for sound rendering (the OSC server); a laptop running the Processing sketch *spAAce*, placed near the center of the WFS system and connected to the server via Ethernet (the OSC client); a Leap Motion and a Wacom Cintiq 22" touch standing in the center of the system as a input/output devices for the client.

3.7 Evaluation

To understand if *spAAce* is correctly abiding to our user's needs, we have tested the prototype of our system in two consecutive phases: an iterative phase followed by a final one. The goals of the iterative testing were to gage product usability and evaluate the features we implemented, so that we could converge and focus only on the main ones. The goal of the final test was to provide a general and more complete evaluation of our system and to show some concrete results. We focussed on investigating subjective qualities inherent to the musical experience, such as enjoyment, expressiveness and perceived affordances, both from the *performer* and the *audience* side. In this paper we focus on the performer's side.

In total we had nine participants for the *performer* testing (all students of Aalborg University). This participant number does not provide statically significant evidence, however it is a reasonable number of participants for evaluating the overall system prototype and retrieve some useful feedback and comments. Seven of the subjects were male, and 6 of them had previous musical experience. Four of them did not have any previous experience with similar softwares, and the rest reported to have some degree of experience. The age of the students was between **19 and 29** years old. Sixteen graduated students from Aalborg University in Copenhagen voluntarily participated in the *audience* test which, however, is not presented in this report. The experiments have been conducted the in the *Multi-Sensory* Laboratory of the Aalborg University in Copenhagen, and we used the same setup described in 3.6. To collect feedback from the participants a survey with qualitative questions has been used and we applied a 7-point Likert scale.

Testing the performers

To evaluate all the performer-related parameters, we designed a two-parts test. The first part aims to evaluate the system and the interface exploration and learnability. For this test the participants were divided randomly in two groups, one with a small training, the other without. The groups were asked to perform some basic tasks to explore the main function of the system. Thus, assuming that all participants achieve the same basic knowledge after completing the first tasks, participants were asked to perform their own creative spatial composition as second part of the test. There were no time constraints for the testing and after the completion of the every tasks, users were asked to fill out a *user evaluation* survey.

3.7.1 Results

According to the data measured, the GUI has revealed to be easy an intuitive to use (data collected show an optimal score with a mean of 5.88 and a low deviation of 1.05). However this result might not be strongly reliable since not enough participants were tested and some of them knew the application on beforehand. Figure 2 shows the overall system experience regarding user usability.

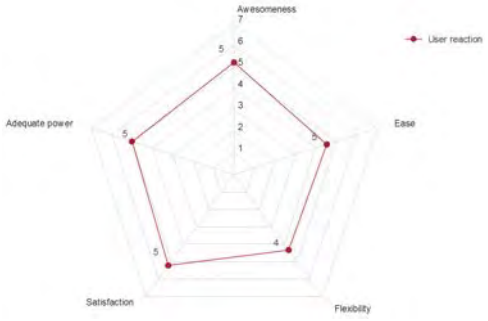


Figure 2. Radar chart that visualises the mean of the 1-7 Likert scale results.

Evaluating the performer experience

Here we tested if the user-friendly interface and the natural approach of the Wacom tablet provides an easy and simple interaction. The answer is basically yes, but on the other hand, Leap Motion was found by all users difficult to manage and would probably need practice in order to be an effective control interface. Figures 3 and 4 show how the users rated the learning curve. The main conclusion is that the system has almost no entry-fee thanks to the already well known tablet interaction, designed and developed with a user-centered framework. Of course the issues with the Leap Motion remain, but research shows that training and improvement of gestures and mapping could lead to a good mastering of the system. Also, comments from the test subjects show that people found Leap Motion quite useful and natural in order to control sound sources, even if the experience was harsh and frustrating at first.

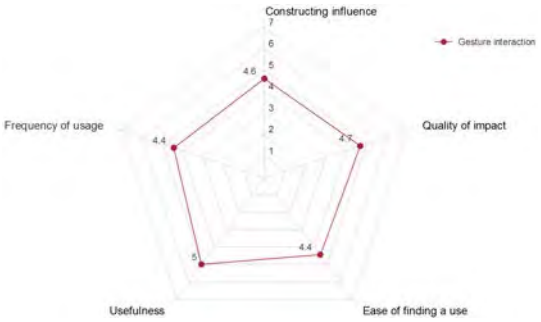


Figure 3. Mean of the Likert scores for evaluating the gesture interaction between the performer and the system.

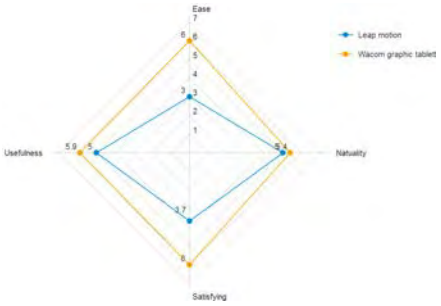


Figure 4. Mean of the Likert scores for evaluating the input devices.

Conducting vs Playing

According to the t-test *p*-value (0.36, alpha = 0.05) there is a significant difference between spatial music trained people and the others. The results show how participants thought that they were mixing rather than playing with the sound sources. We believe the title of the question was probably misunderstood. The ”mixing” rated values belonged to participants highly trained in WFS techniques and/or amateur composers, while the participants who rated ”playing” were less trained in these regards. There is perhaps an expectation bias and also a problem of terminology which could be improved in further tests.

Satisfaction and Enjoyability

As show in figures 3, 4 and 2 the satisfaction was very high, and a very positive feedback, ofter even passionate, was felt with respect of the subjects enjoyability and satisfaction. This shows that *spAAce* can be very enjoyable, even if the sound sources movements are not fully understood. However we think that the novelty of the system and the astonishment of watching the system in the laboratory space, has influenced greatly the subjects.

4. WFS GAME

4.1 Introduction

One goal of this project is to investigate how convolution reverb affects people’s perception of distance in a wave-field synthesis setup environment. In order to achieve this, an auditory game prototype has been developed and to keep the focus on auditory perception, players do play the game

blindfolded. The style of the game is horror/survival and the user is exposed to several "enemies", which he/she needs to localize and eliminate by using a Nintendo WiiMote game controller to "throw" sounding objects towards them. There are three types of enemies with different mechanics and sonic characteristics that will be described in a next section. They all are created by using point sources and focused sources, and they are wither static or moving around or towards the player after they appeared in the virtual space around the player.

4.2 Impulse Response Reverb

The environment of the game resembles a commercial ship, thus a background ambience soundscape was designed containing sounds such as an air fan, water drops from a broken pipe, wind sound coming from outside the ship and rat squeaks. Acquiring the impulse response from a ship was essential, since this project relies on investigating the role of convolution reverb in distance perception for WFS. The Impulse Response was captured using the ESS (Exponential Sine Sweep) method [32], in a big metal ship owned by the Illutron Collaborative Interactive Art Studio [1] . The following equipment was used in the process: a MacBook Air Laptop, a Dynaudio BM5 MK I speaker, a Rode NT2 omnidirectional microphone and a Focusrite Scarlett 8i6 audio interface. The recording and deconvolution was handled via the Apple Logic Pro X internal Impulse Response Utility.

4.3 Hardware and Software

Since the game was designed to allow the player to move freely in the area inside the array of loudspeakers, a shooting system has been implemented coupling a WiiMote with two motion capture markers captured by an array of 16 OptiTrack Flex 3 infra-red cameras. Of the two MoCap markers one had been placed on the player's shoulder and another one on the WiiMote.

Unity3D was running as a local debug software as well as an interpreter from VPRN to OSC, since NaturalPoint cannot send OSC data. The Unity5 game engine, the WiiMote OSCulator 2.13.3 receiver and the WFSCollider sound engine software were all running on a separate Mac Pro computer (dual Intel Xenon 12 core processor, 64 GB DDR3 RAM).

The WFS audio stream is delivered from an RME MADIface USB interface via two DirectOut ANDIAMO 2 MADI converters, each connected to 32 M-Audio BX5 D2 loudspeakers. In total the WFS system delivers sound trough 64 loudspeakers aligned one to the other and calibrated in their output level. The WFS system consists of 4 arrays of 16 loudspeakers each, displaced to form a square of 4 by 4 meters inside which users can freely move.

The wavefield synthesis had to happen in realtime after the user input and according to the enemies positions, to maintain the desired playability. Also for this project the choice of WFS engine fell on WFSCollider, the audio spatialization engine for Super Collider developed by Wouter Snoei at The Game of Life Foundation [2] . Beside the capability of rendering wavefield sound, WFSCollider also serves as an intuitive digital audio workstation (DAW) offering functionalities such as multi-track mixing, effect chains, auxiliary buses, featuring also an easy OSC control on every parameter, thus making it very suitable for the desired setup of this work. In WFSCollider sound sources are triggered and controlled in position and properties by control messages coming from Unity5 and OSCulator.

4.4 Sound Design

Three types of enemies were designed for the game, which will be described as Enemy 1, Enemy 2 and Enemy 3. The numbering represents the order of apparition. All these enemies spawn randomly at different locations, from three different "rings" or levels of distance. Enemy number one will always appear from the further area, while enemy number two will be appearing from the closest one, leaving enemy number three to appear from the mid one. See Figure 5 where E1, E2, E3 represents the three enemies; the rings represent the three different areas of distance where enemies are coming from; the square represents the physical space enclosed by the WFS array, and P represents a player.

The lifetime of each enemy is 1 minute. This limit is implemented to compensate for an issue encountered in the pilot tests: sometimes, the user cannot hit an enemy.

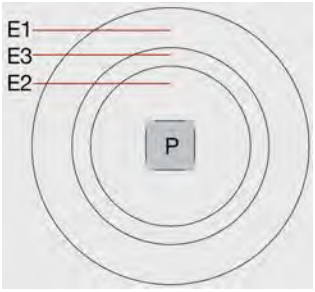


Figure 5. Map of the virtual space.

1. Enemy 1 slowly moves on a linear trajectory towards the player and tries to "hit" him/her, emitting a continuos flow of sound while it moves. E1 depicts a human dragging a heavy metal object and the most evident sound characteristic of this enemy is its slow footstep movement. Several sounds are used to create it: a pair of foot sounds alternating, a recorded heavy breathing sound, as well as the sound of a metal object dragged on a metal surface. All sounds are grouped together; as long as their virtual position is located outside the ring of speakers they are rendered as point sources, and when they get close to the player and "enter" the loudspeaker area, they become focused sources.

2. Enemy 2 position is static. E2 represents a woman who is breathing fast and sobbing while spinning a

[1] http://illutron.dk

[2] http://gameoflife.nl

chain, its sound characteristics are then female screams and a swinging flail weapon sound. This enemy is immobile and it alternates short silences and sounds. Three sounds were used to create her, a chain links clinker, a recorded sobbing/ breathing sound and a vocal sound. Just as Enemy 1, these sounds are grouped together and if the enemy appears in the area behind the loudspeakers they are rendered as point sources, otherwise they are rendered as focused sources if E2 appears in the area inside the loudspeakers array.

3. Enemy 3 combines together some of the mechanics and sound characteristics of E1 and E2. Its position is static but every 20 seconds it spawns a series of moving distractive sounds, which travel around the virtual space where the player is, making it harder of the player to locate and eliminate him/her. E3 symbolises a ward drum player, with a twist, and only one sound source is used to create it: a rhythmical uninterrupted drum loop. For the distraction sound, several male exhale sounds were used, being processed to sound like a wrath. These two sounds (E3 and its "distractors") combine together one continuous sound cue with a series of short sounds that appear and go. Just as the other two enemies, all the sounds are point sources when their virtual location is located outside the speaker area, and focused source otherwise.

4.5 Test Design

Each subject is introduced to the game mechanics by going through a training phase which consists of three stages, each lasting one and a half minutes and dedicated to set the player familiar with the relation between the gesture he/she has to perform (direction and force of the gesture) to "throw" a sound against an enemy, and the distance at which the sound is thrown. In this phase the subject is already blind-folded and is requested to locate and hit the virtual sound by "shooting" another sound with the WiiMote towards it, according to the subject's perception of how far the target sound is located.

The training sound to hit resembles a synthetic metronome beat and is located into one of the three circular areas visible in Figure 5; the player receives a sound feedback to understand if the shoot was good (the gesture was performed with the exact force needed to launch the sound into the desired area) or not. The sound to hit remains the same on all three stages but the distance increases from the inner to the outer circular areas as the stages progress. Once a participant has been familiarised with how the game works, the actual testing starts.

The real game/test comprises also of three stages, one for each of the three enemies. In each of the three stages the participant is exposed to eight instances of every enemy, four of these are presented with impulse response reverberation and four without, randomly assigned. The participants actions are tracked throughout the test and logged to files. The log entries include player position, collisions coordinates and timing, number of shots fired during each of the phases and number of enemies spawned and hit.

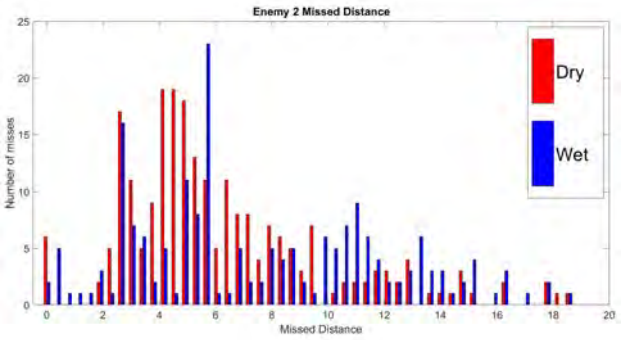


Figure 6. "Spatial precision": missed projectile distance.

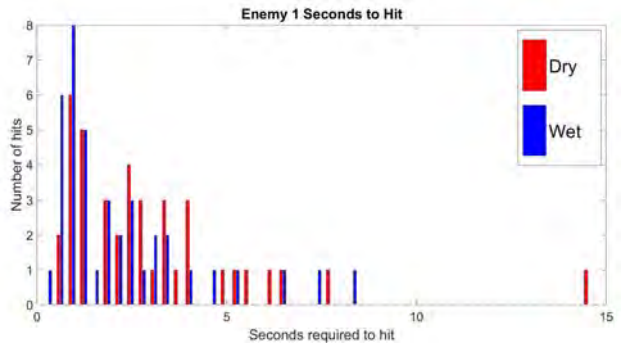


Figure 7. "Game performance": time required to hit an enemy.

4.6 Experiment Results

The test was performed on 9 males and 1 female participants aged between 21 and 27, all of them reporting to have musical training and suffer no hearing loss. The results have been gathered in three main categories. These include a distance between the projectile impact position and a target enemy (namely "spatial precision"), the time required to correctly shoot an enemy and the total accuracy of shots (number of good shots versus bad shots). In each of these categories a paired t-test has been performed to verify whether or not the presence of reverberation (wet/dry parameter) had an influence on the participants' achieved scores. This procedure was carried out three times, once for each of the enemies that the test participants were exposed to, thus getting a total of 9 tests. The paired t-test revealed that results provide no statistical significance required to determine whether or not entries related to the reverberant environment condition differ from those logged in the dry condition (= with no impulse response based reverberation - only the natural dry reverberation provided by the lab room in which the WFS system is placed). Among all the nine tests, only one yielded significant difference between two conditions - the one ran on missed projectile distance entries during an Enemy 2 phase (p=0,046). However, even in this case, the difference between mean values is equal to 20.1949-16.5106=3.6843, a relatively low number -Figure 6.

Statistical analysis of data does not bring a solid answer to the hypothesis that a difference is in place between the performance achieved in shooting at the correct distance

in wet or dry reverberation conditions. Also the analysis performed on the number of seconds required to hit an enemy, shows no difference in all nine cases, so only one plot is here presented as an example of the results (Figure 7.), leaving further reflections to the discussion part.

4.7 WFS Game Discussion

The analysis of data shows no significant difference in the results performed with and without convolution reverberation, nevertheless it is worth mentioning that both the system used as a tool to perform the test, and the experiment design itself have possibly affected the outcome substantially. First of all, the gestural interface was commonly reported by subjects to be counter-intuitive and non-reliable and hence it can be partially blamed for an overall poor performance of the users (in terms of accuracy, time required to aim and average missing distance); moreover, this aspect raised frustration and distraction from the task. Consequently, participants tended to become tired towards the end of the test, which led to further deterioration of their score. The main reason behind this issue has been addressed as the delay between the motion capture system and the WiiMote input data flows. The stream of data from the MoCap computer, to the computer receiving the WiiMote data, is affected by a small lag, that causes incorrect reading on the users hand position in the moment when they trigger the WiiMote button to "throw" their sonic weapon. This small lag sometimes causes a wrong reading of the relative position of the two markers (the one placed on the player's shoulder and on the WiiMote), which in the end can generate a wrong shooting angle. This error is more pronounced in users who perform a very fast and energetic movement with the WiiMote. This problem could be overcame by changing the shooting mechanism. Another solution to overcome the lag would be to redesign the data flow either making use of a single computer, or relying only on the WiiMote internal sensor data fusion to generate an accurate shooting direction.

The evaluation aspect of this project was revolving about the impact of convolution reverb in a WFS system, but this is not the only way to create artificial room simulations; different techniques could be adopted instead of a direct convolution of the sound sources: for future studies another option could be to model reverberation as four planar waves representing physical walls and fed with all the signals to be convolved. Also incorporating completely different approaches, such as Schroeder reverberators might be worth investigating. It is in the end worth mentioning that this project completely omits the proprioception aspect of the experience. Early tests suggest that the perception of the shooting hand might influence the shooting performance from player to player. A further experiment investigating this aspect could provide useful information in understanding the analysed data, as well as provide useful knowledge for designing interactions and interfaces for alike systems. At last, also sound design aspects could be affecting the results and be worth investigating more, since besides comparing moving sounds and static sounds, the sounds themselves embed different temporal and spectral contents which might affect subject's perception.

5. CONCLUSIONS

We presented two preliminary studies of gesture control of Wavefield Synthesis performed in our research group: a graphical application called *spAAce* which let users to control real-time movements of sound source by drawing trajectories, and an acoustic-based game which aimed to investigate the impact of convolution reverberation over the perception of distance in a wavefield synthesis scenario.

5.1 spAAce

The first prototype of this application has been developed bound with WFSCollider, an open-source software based on Supercollider. In order to communicate with the software, Open Sound Control protocol has been employed. The *spAAce* application has been implemented using Processing, a programming language for sketches and prototypes within the context of visual arts. This application aims to create a new way of interaction for live performance of spatial composition and live electronics. Promising results have been found in the small test performed, encouraging the authors to further implement the system. Future work will focus on a more extended test (to eliminate possible biasing caused to the users by the novelty of the "experience") and on further development of the available tools used to create and control sound trajectories. Also, a more general version of the *spAAce* software is desired to be developed in a later stage, to run the software on iOS and Android tablets.

5.2 WFS GAME

While the test results for this project were mainly inconclusive, the miss distance for Enemy 2 was statistically proven to be influenced by the reverb status, accepting the null hypothesis, indicating that a dry sound sources were slightly easier for the participants to hit. Nevertheless, the platform used for this research is worth further development as it provides more possibilities for examining embodied interaction in a virtual auditory environment. Also, besides the considerations on further possibilities of study on how to implement a more effective setup for the experiment purposes, another interesting way of exploiting it would be to include interaction between people, so that more users can interact with the environment.

6. REFERENCES

[1] F. Rumsey, *Spatial audio*. CRC Press, 2012.

[2] W. Snow, "Basic principles of stereophonic sound," *Audio, IRE Transactions on*, no. 2, pp. 42–53, 1955.

[3] B. François, "Pour une musique invisible: un acousmonium," *Festival International du Son Haute Fidélité Stéréophonique*, pp. 125–134, 1975.

[4] J. M. Chowning, "The simulation of moving sound sources," *Journal of the Audio Engineering Society*, vol. 19, no. 1, pp. 2–6, 1971.

[5] I. Recommendation, "775-1, multichannel stereophonic sound system with and without accompanying picture," *International Telecommunication Union, Geneva, Switzerland*, 1994.

[6] V. Pulkki, "Virtual sound source positioning using vector base amplitude panning," *Journal of the Audio Engineering Society*, vol. 45, no. 6, pp. 456–466, 1997.

[7] ——, "Generic panning tools for max/msp," in *Proceedings of International Computer Music Conference*, 2000, pp. 304–307.

[8] T. Lossius, P. Baltazar, and T. de la Hogue, *DBAP–distance-based amplitude panning*. Ann Arbor, MI: MPublishing, University of Michigan Library, 2009.

[9] N. Peters, T. Matthews, J. Braasch, and S. McAdams, "Spatial sound rendering in max/msp with vimic," in *Proceedings of the 2008 International Computer Music Conference*, 2008.

[10] M. A. Gerzon, "Ambisonics in multichannel broadcasting and video," in *Audio Engineering Society Convention 74*. Audio Engineering Society, 1983.

[11] M. Gerzon, "Surround-sound psychoacoustics: Criteria for the design of matrix and descrete surround-sound systems." *Wireless World. Reprinted in An anthology of articles on spatial sound techniques, part 2: Multichannel audio technologies. Edited by F. Rumsey. Audio engineering society, Inc, 2006.*, 1974.

[12] A. J. Berkhout, "A holographic approach to acoustic control," *Journal of the audio engineering society*, vol. 36, no. 12, pp. 977–995, 1988.

[13] A. J. Berkhout, D. de Vries, and P. Vogel, "Acoustic control by wave field synthesis," *The Journal of the Acoustical Society of America*, vol. 93, no. 5, pp. 2764–2778, 1993.

[14] S. Brix, F. Melchior, T. Roder, S. Wabnik, and C. Riegel, "Authoring systems for wave field synthesis content production," in *Audio Engineering Society Convention 115*. Audio Engineering Society, 2003.

[15] E. Corteel and T. Caulkins, "Sound scene creation and manipulation using wave field synthesis," *Rapport technique, IRCAM*, 2004.

[16] R. Pellegrini and C. Kuhn, "Wave field synthesis: Mixing and mastering tools for digital audio workstations," in *Audio Engineering Society Convention 116*. Audio Engineering Society, 2004.

[17] J. Ahrens, M. Geier, and S. Spors, "The soundscape renderer: A unified spatial audio reproduction framework for arbitrary rendering methods," in *Audio Engineering Society Convention 124*. Audio Engineering Society, 2008.

[18] M. Baalman, "Application of wave field synthesis in the composition of electronic music," in *International Computer Music Conference, Singapore*, 2003, pp. 1–4.

[19] M. A. Baalman, "Updates of the wonder software interface for using wave field synthesis," *LAC2005 Proceedings*, p. 69, 2005.

[20] W. Snoei. Wfscollider 2.2.1. [Online]. Available: https://github.com/GameOfLife/WFSCollider

[21] T. Sporer, "Wave field synthesis-generation and reproduction of natural sound environments," in *7th International Conference on Digital Audio Effects (DAFx-04), Naples, Italy*, 2004.

[22] S. Spors, R. Rabenstein, and J. Ahrens, "The theory of wave field synthesis revisited," in *124th AES Convention*, 2008, pp. 17–20.

[23] E. N. G. Verheijen, "Sound reproduction by wave field synthesis," Ph.D. dissertation, TU Delft, Delft University of Technology, 1998.

[24] S. Spors and R. Rabenstein, "Spatial aliasing artifacts produced by linear and circular loudspeaker arrays used for wave field synthesis," in *120th AES Convention*. Citeseer, 2006.

[25] K. Stockhausen, "Musik im raum," in *La Rassegna Musicale: Problemi della musica, oggi (Numero Speciale)*, G. M. Gatti, Ed. Giulio Einaudi editore, 1961, vol. 31, IV, pp. 397–417.

[26] J.-M. Jot and O. Warusfel, "Spat: a spatial processor for musicians and sound engineers," in *CIARM: International Conference on Acoustics and Musical Research*, 1995.

[27] "CodeOfLife wfscollider," https://github.com/GameOfLife/WFSCollider.

[28] M. Geier, T. Hohn, and S. Spors, "An Open Source C++ Framework for Multithreaded Realtime Multichannel Audio Applications," in *Linux Audio Conference*, Stanford, USA, Apr. 2012. [Online]. Available: http://lac.linuxaudio.org/2012/download/lac2012_proceedings.pdf

[29] X. Favory, J. Garcia, and J. Bresson, "Trajectoires: une application mobile pour le contrôle et l'écriture de la spatialisation sonore," in *27ème conférence francophone sur l'Interaction Homme-Machine.* ACM, 2015, p. a5.

[30] R. Penha and J. Oliveira, "Spatium, tools for sound spatialization," in *Proceedings of the Sound and Music Computing Conference*, 2013.

[31] "Processing," https://processing.org/.

[32] A. Farina, "Advancements in impulse response measurements by sine sweeps," in *Audio Engineering Society Convention 122*. Audio Engineering Society, 2007.

INTERFACES FOR SOUND: REPRESENTING MATERIAL IN POP MUSIC PRODUCTIONS

Florian Grote
Native Instruments GmbH
Berlin, Germany
florian.grote@native-instruments.de

ABSTRACT

Sound is the foundation of music composition in contemporary popular cultures. As the medium of popular music, sound is an important dimension in which artists establish their identities and become recognizable. This presents a radical departure from the focus on written notation and a pre-set corpus of instrument timbres found in classical Western-European music. To create in the medium of sound, contemporary composers utilise digital production systems with new interfaces, many of which are built upon the waveform representation as their cornerstone. This waveform representation is an interesting bridge between the analog world, from where it borrows its appearance as a seemingly continuous line, and the digital world in which it exists as a visualisation of a digital model describing continuous audio material in discrete sample points. This opens up possibilities to augment the waveform representation with interactions for algorithmic transformations of the audio material. The paper investigates the cultural implications of such interfaces and provides an outlook into their possible futures.

1. INTRODUCTION

Sound is the main component of contemporary music composition and production in popular styles. Whereas throughout the last centuries of Western music history, the temporal arrangement of notes and the timbre with which they became audible were separate considerations, these two realms have folded into one in most of pop music in the late 20th and early 21st century. Thus, contemporary music composition has firmly entered the era of sound. However, the realm of music composition technology has not yet fully followed suit, and instead remains centred around representational paradigms of the pre-sound era, as will be demonstrated. Composers and performers have found ways to deal with this situation, but the lack of actual representations of sound aspects in the creative environment may well be a dampening factor especially in the development of new musical creative talent.

In this paper, I will first investigate cognitive-cultural aspects of sound and in the second part analyze current representation forms in successful offerings of music technology.

2. SOUND IN CULTURE

In many ways, sound is not a new phenomenon or even cultural discovery. Traditional music cultures in Asian as well as African traditions have relied on sound all along [5, 11]. It was mainly in Western music cultures of the last centuries that note and timbre became separate aspects [4]. Notation was the driving force behind this separation, itself of course being a tool to enhance division of labor between a composer and the performer interpreting a written composition, giving it a voice in the form of expressive timbre. The boundaries were not that clear from the beginning, and we can find numerous attempts to exert tight control on timbre from a score, as well as attempts to free the interpretation by the instrumentalists from the shackles of an all to clear written composition. Non-Western music traditions have typically not relied on written notation for their compositions, but instead have nurtured canonization through collaborative practice, i.e., imitation and oral tradition. The resurgence of sound in contemporary global popular music cultures has led to a marginalisation of written notation, giving rise to alternative techniques and strategies that seem more appropriate for the needs of composers.

Culture in general and global pop culture in particular is driven by the dynamics of artist identities and their influence on the publics they reach. Popular artists define their way of doing things, broadcast by mass media and disseminated via social media, and this is then reacted on by those who feel addressed. In the past as well as the present, this has spawned fashion trends in clothing and hair cuts, and generated a wide variety of imitators. Next to the visual dimension of popular culture, it is the sound of the music productions by those artists that carries their identity in making them unique, yet at the same time promising the possibility to imitate. Sound and visual aspects combine to form a strong concept that can be recognised in all forms of communication. When one recognises a certain song or a piece of music as the work of a particular artist, the overall sound of the recorded or performed music often plays a central role. Sound as a concept is amazingly context-invariant, as Wicke points out [16], meaning that recognition of an artist or style works even when perceived in different spaces, on different devices, or in varying social contexts, where the actual frequency pattern of the acoustic phenomenon can be very different. It appears that sound is a recognisable entity that is greater than the sum of its parts, mainly being note frequency and duration as well as timbre. Sound

transcends those individual aspects and has become an overarching, unifying phenomenon that serves as a great source of identity for artists operating within popular cultures.

3. SOUND IN MUSIC PRODUCTION

The ephemeral, yet clearly recognisable character of sound as a phenomenon presents a considerable challenge for music production technology. After all, if the medium of recorded and performed music in popular cultures is sound rather than harmonic structures or timbre, then compositions have to be written in this medium. How, then, is this medium represented in today's instruments for music production, and which strategies can be identified for developing future representations?

The process of music production is usually hidden away from public visibility, but the duality of sound and visual representation is present there as well, albeit on another level, and has to be dealt with by the creators themselves. The semantic structure of a digital music production system is still very closely related to the technical setups of the analog world. The notion of signals flowing through an audio path is defining much of the layout of software interfaces, with individual elements of the interface mimicking the layout and the look of analog devices such as mixer boards, effects units, and recorders. On the instrument side, the picture looks similar. Analog synthesizers are often modelled in software form, with their interfaces being skeuomorphic representations of the original analog devices. The only truly digital representation of musical information found in music software are usually the note editors, which fit the bill of revealing information in discrete and numbered steps using a limited language of symbols. However, the typical representation style of the piano roll has that name because it has been used for the rolls holding the compositions for player pianos since the late 19th century, a technology which itself built upon barrel organs that had been available some 200 years earlier [7, 8]. Thus, this genuine digital representation has a history that is much longer than that of the music software it is a part of.

Another representation is an interesting mix of analog and digital: The waveform display and the tools offered to interact with it. Essentially, the waveform display is giving the false impression of being the representation of an analog continuum, by showing the form of a continuous wave. A very similar waveform would be drawn if one would measure the current flowing to the loudspeaker at the end of the musical signal chain, or actually the movements of the loudspeaker cone exciting changes in air pressure. However, on the computer screen this seemingly continuous line is drawn by individual pixels, and in a similar way, the corresponding signal is in reality only described by a finite number of individual samples, with no information as to what is in between. What makes this representation interesting is that, although it is rooted in the analog domain and could be produced there as an image, it becomes the foundation for new interactions that are only possible in the digital world. Here, algorithms can be devised that generate or extract information from the description to allow for interactions that would previously have required a human listener to be conceived of. These interactions with the description of the signal can be described as reproduction media turning into musical instruments, as Großmann has pointed out [3].

4. INTERFACES

In actual systems, we encounter a mixture of note-based and waveform-centric interfaces. For example, in a production software, or digital audio workstation (DAW) like Steinberg Cubase, we see both note-based and waveform-based tracks, operating with different data formats, MIDI for notes and audio files for waveforms. These are the classic representations of these data formats, and they also represent the aforementioned traditional "Western" separation between notes and timbre.

Figure 1. Note-based and waveform tracks in Cubase.

Representations such as those shown above are the baseline standard in most current music production systems. However, they have often been augmented with additional interactions that try to add more sound control to the editing of notes on the one hand, and on the other bring event-type editing control to waveforms.

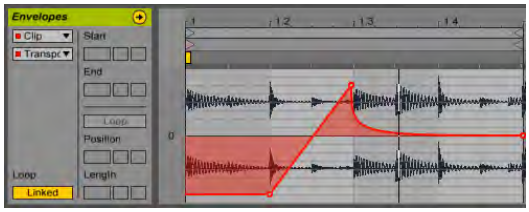


Figure 2. An envelope that controls a sound parameter in direct relation to the audio waveform in Ableton Live.

Figure 3. A waveform that has been split into its tonal components and mapped to a piano roll representation in Melodyne DNA [10].

There are examples of systems that go even further than this. The first would be the DJ software Traktor DJ for iOS, which established a paradigm of playing the waveform directly with the fingers, similar to how notes would be entered traditionally.

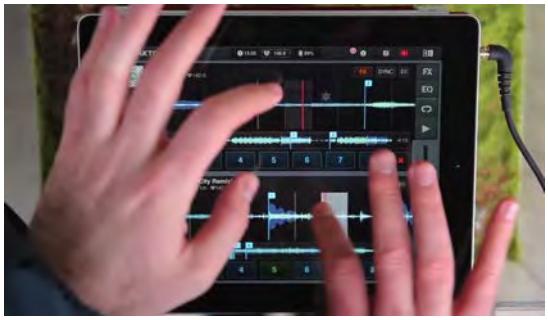


Figure 4. DJ Shiftee playing Traktor DJ in "Freeze Mode", where the waveforms have regions that can be played in realtime by touching them [6].

It should not come as a surprise that utilising waveforms as the representation of the audio material for performance purposes is found in DJ applications like Traktor DJ. After all, DJing as a cultural and artistic practice has been built on the foundation of dealing with recorded music in its phonographic form as material in the sense described by Großmann and Hanáček [2]. A similar tradition can be found in the sampling practices of hip hop, where devices like Akai's MPC are offering features to slice waveforms into small regions, which can then be played directly from pads on the hardware.

Figure 5. Akai MPC 500 with "Chop Shop" waveform slicing activated. Regions of the waveform seen in the display can be played back directly from the grey pads in the middle of the device [9].

On the desktop, DJ applications like Serato DJ or Traktor Pro have another enhanced waveform representation, where the energy of different frequency ranges within the audio material is visualised together with its absolute level. This makes sense especially for many electronic music genres, where it becomes possible to identify parts of individual instruments, e.g. see when a hi-hat line comes in or when a bass line switches to a different pattern [1, 12].

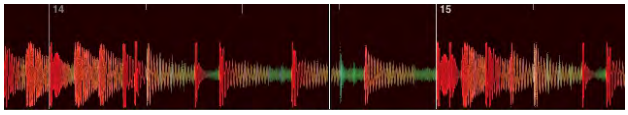


Figure 6. A multi-coloured waveform in Serato DJ.

Another example that greatly expands the waveform-based representation of sound is the "Drummer" feature in Apple's Logic Pro X production system. Here, a cultural context can be invoked by selecting a drummer-persona, which then makes available a prototypical variety of drum grooves supposedly expected from a drummer in this context. This high-level representation is interactive, as the user can change parameters like loud/soft or simple/complex, set complexity ranges for individual instruments, and also set a "Humanize" factor. This in itself is an interesting new representation, as it incorporates cultural semantics in an algorithmic rhythm generator, but it goes further by rendering the results of the rhythm generator as both note and waveform material into the main production environment. There, it can be subjected to further transformations in either domain, thus creating a highly flexible, integrated means of handling sonic material in an abstracted form.

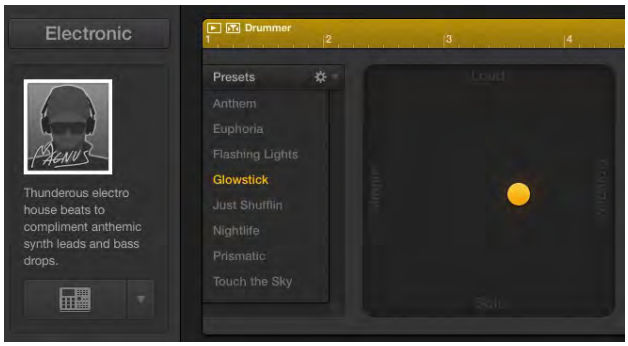


Figure 7. The "Drummer" instrument in Apple's Logic Pro X, with the persona "Magnus" loaded [13].

The "Drummer" in Apple's Logic music production system presents a far-reaching abstraction in the direction of representing sound by integrating high-level semantics from music practice with the more technological interfaces of interactive notation and waveforms. Even so, its approach also highlights the limitations of a design that tries to mediate between a simplistic approach of mapping various rhythmic variations onto four dimensions of parametrical freedom and the flexibility of full editing of the generated note material and its result in the audio domain. The cultural semantics used in this design are highly stereotypical, up to an outline of the drummer persona's head with visual cues such as headphones and sunglasses for the "Magnus" persona in the "Electronic" genre. Practices in popular cultures are well-versed in methods of cultural sampling, where semantics or even stereotypes are re-contextualised in other genres. However, whether a reductionist approach such as the Logic "Drummer" will indeed be taken seriously in established cultural practices remains questionable, at the very least.

5. CONCLUSIONS

The interactive waveform representation, with its additional enhancements, has become a mainstay in interfaces for music production in contemporary popular culture. Here, the phonographic material becomes accessible to instrumental practice and production techniques that would be impossible without digital technology, where audio material is sampled and thus laid open to algorithmic processing. Defining the degrees of freedom offered by this algorithmic processing by means of cultural semantics is an interesting and promising field, which is still in its infancy. The challenge here is that both realms have to be taken seriously not only in the technological configuration, but also in the design and presentation of interfaces for users. Notation editors and interactive waveform views are well-established instruments in the practice of electronic music genres, but so is the method of cultural sampling. In this latter area, definitions of styles and genres have been developed into a highly differentiated and dynamic field of cultural semantics, and their appropriate usage is paramount to the success of a technology being credible and thus acceptable for creative practices in popular cultures. Reductionist approaches may be adequate for the sake of easy adoption, but only when they are the result of adaptive fine tuning, taking the cultural context of its use cases into account. The level of differentiation in the offer will then be reflected in the depth of cultural specification in the contexts of its use.

6. REFERENCES

[1] E. Golden, “Reading Wave Forms”, in DJ TechTools, January 5, 2010 [Online]. Available: http://djtechtools.com/2010/01/05/understand-your-waveforms/ [Accessed: April 11, 2016].

[2] R. Großmann and M. Hanáček, “Sound as Musical Material”, in Sound as Popular Culture: A Research Companion, J. G. Papenburg and H. Schulze, Eds., MIT Press, 2016, pp. 53–64.

[3] R. Großmann, “Distanzierte Verhältnisse? Zur Musikinstrumentalisierung der Reproduktionsmedien”, in Klang ohne Körper: Spuren und Potenziale des Körpers in der elektronischen Musik, M. Harenberg and D. Weissberg, Eds., transcript, 2010, pp. 183–200.

[4] R. Großmann, “Phonographic Work”, in Sound as Popular Culture: A Research Companion, J. G. Papenburg and H. Schulze, Eds., MIT Press, 2016, pp. 355–366.

[5] M. J. Kartomi, R. A. Sutton, E. Suanda, S. Williams, and D. Harnish, “Indonesia”, in The Garland Handbook of Southeast Asian Music, Routledge, 2011, pp. 334–405.

[6] Native Instruments, “DJ Shiftee Takes On TRAKTOR DJ”, 2013 [Online]. Available: https://www.youtube.com/watch?v=oFPTyZoxO3Q [Accessed: 11-Apr-2016].

[7] A. W. J. G. Ord-Hume, “Player Piano: The History of the Mechanical Piano and how to Repair it”, Allen & Unwin, 1970.

[8] A. W. J. G. Ord-Hume, “Automatic Organs: A Guide to Orchestrions, Barrel Organs, Fairground, Dancehall & Street Organs Including Organettes”, Schiffer Pub., 2007.

[9] C. San Segundo, “Akai MPC 5000”, delamar.de, January 18, 2008 [Online]. Available: http://www.delamar.de/musik-equipment/akai-mpc-5000-1083/ [Accessed: April 13, 2016].

[10] M. Senior, “Celemony Melodyne DNA Editor” [Online]. Available: http://www.soundonsound.com/sos/dec09/articles/melodynedna.htm [Accessed: April 11, 2016].

[11] R. M. Stone, “Exploring African Music”, in The Garland Handbook of African Music, Garland Publishing, 2000, pp. 13–22.

[12] M. Strauss, “Review: Native Instruments - Traktor Pro 2”, in Resident Advisor [Online]. Available: http://www.residentadvisor.net/review-view.aspxid=8962 [Accessed: April 11, 2016].

[13] M. Wherry, “Apple Logic Pro X” [Online]. Available: http://www.soundonsound.com/sos/sep13/articles/pro-x.htm [Accessed: 11-Apr-2016].

[14] M. Wherry, “Product Review - Steinberg Cubase 8.5” [Online]. Available: http://www.soundonsound.com/sos/apr16/articles/cubase85.htm [Accessed: April 11, 2016].

[15] D. White, “Video: Shiftee Rocks Traktor DJ; Freeze Mode Mastery”, in DJ TechTools, April 4, 2013 [Online]. Available: http://djtechtools.com/2013/04/04/video-shiftee-rocks-traktor-dj-freeze-mode-mastery/ [Accessed: April 11, 2016].

[16] P. Wicke, “The Sonic”, in Sound as Popular Culture: A Research Companion, J. G. Papenburg and H. Schulze, Eds., MIT Press, 2016, pp. 23–30.

DEVELOPING A PARAMETRIC SPATIAL DESIGN FRAMEWORK FOR DIGITAL DRUMMING

Jeremy J Ham
RMIT University
jeremy@surfcoastarchitecture.com.au

Daniel Prohasky
RMIT University
daniel.prohasky@rmit.edu.au

ABSTRACT

This research operates at the intersection of music and spatial design within the context of improvised digital drumming. We outline a creative design research project founded on a series of affordance experiments that explore the ways in which the tools of spatial design can inform understandings of 'referent (Pressing 1987)' improvised patterns and phrases employed by experienced drummers. We outline the stages and process of development of a parametric computational framework using software from the spatial design industry to provide affordance (Gibson 1979) to understanding the complexities of drum improvisation.

The 'ImprovSpace' Grasshopper script, operating within Rhino3D™ enables the 3D spatialization of digital drum-based improvisations wherein the parameters of drum notes, duration and velocity all can be flexibly manipulated. Drum phrases and patterns can be compared individually and clusters of repeated elements can be found within a larger corpus of improvisations. The framework enables insights into the specific attributes that constitute individual style including micro-timing, rubato and other elements of style. It is proposed that, by bringing these improvisations into visual and spatial domain in plan, elevation and isometric projections, a theoretic musico-perspectival hinge may be deconstructed that provides insights for visually and spatially dominant musicians within reflective, educational and other contexts.

1. INTRODUCTION

This paper reports on early PhD project work founded on the practice of creative digital drumming as examined through a lens of spatial design. This research centres on the first author's creative practice as an architect and improvising drummer. Creative drumming, in this context, is the advanced playing of the drum kit with the express intention of exploring boundaries of timing, polyrhythm and space across solo and group contexts. Positioning the research on the intersection of the disciplines of music and spatial design allows theoretical, technical, representational and computational concepts and methodologies from one discipline to be used to examine the other. We are primarily interested in the ways in which spatial design tools can add to the body of knowledge in understanding musical improvisation.

The Field of Musical Improvisation (FMI) (Cobussen, Frisk et al. 2010) provides the theoretical and practical context within which creative drumming improvisations are performed. The FMI describes 'the precise progress and structure of an improvisation (as) essentially capricious,' 'between order and disorder, between structure and chaos, between delineation and transgression or extension'. In developing this parametric framework, we are attempting to make finite some of these aspects of the 'infinite' art of improvisation.

There has been a long history of architects attempting to translate music into architectural form and space. Martin (1994), in 'Architecture as a Translation of Music' established a model for the examination of music and architecture on three levels: 'Based on acoustics', 'Instrument as Architecture' and 'Layered Relationships'. Martin describes the 'y-condition' as 'the middle position of music and architecture when translating one to another,' finding an organic union between the two.

The computer thus serves as a good tool to facilitate these translations- particularly in the area of 'layered relationships'. The foundation of many music: spatial design (architecture) translations using the computer is the 'reduction of all information to a binary signal, be it a picture, a text, a space or a sound - all data is recorded as a binary sequence allowing computation as defined by programming languages and communication through networks according to transmission protocols' (Lab-au.com 2015). The principal that 'the byte shall be the sole building material (Levy 2003)' acts to enable compositional opportunities within the spatial dimension,. This also is reductionist, as not all of the properties of music can be translated adequately or completely. Mediating this 'y-condition' computationally requires the 'practiced hand' of the digital craftsperson (McCullough 1998).

MIDI data derived from digital instruments forms the basis for many translations from music into spatial design. Ferschin, Lehner et al. (2001) developed COFFEE™, a language for describing objects in space that extends the functionality of Cinema 4D™ for 3D modelling, rendering and animation. 'The main part of the translation process consists of a mapping of musical parameters, like pitch, duration, tempo, volume, instrument to architectural parameters as shape, size, position, material so that a piece of music in MIDI notation can be translated to a geometric structure containing shapes with materials'. Further parametric translations, framed as 'Spatial Polyphony' are undertaken by Christensen (2008), utilising MIDI musical data. Johan Sebastian Bach's The Well-tempered Clavier fugue is translated

into ASCII text, imported into Microsoft Excel™ as numbers, then parametrically translated into form using CATIA™. Christensen concludes that the process, whilst limited, allows us to freeze the music 'in a single moment allow(ing) one to see the shape of the entire piece simultaneously, something which is not possible when listening to a performance'.

Marcos Novak has, over a long period and through many projects, proposed that these binary data can be crafted in a designerly fashion: 'the question arises: what to do with these data? What is an appropriate poetics for a world such as this?' He proposes 'Archimusic' as 'the conflation of architecture and music' within the realm of cyberspace. He states: 'Archimusic is to visualisation as knowledge is to information', and proposes virtual acoustic displays as the sonification of 'Archimusic'. *(Novak 1992- 2007).*

Architect Jan Henrik Hansen adopts figurative and literal translations in many built works of sculpture (musical sculptures) and building elements, merging music and architecture within private practice (Hansen 2015). Practitioners like Hansen, along with academy-based researchers above prove the potential for deep examinations of 'architecture as frozen music.' We are, however, less interested in translating music into architecture, but using parametric design tools to provide affordance for the understanding of the process of making music. The emphasis is not on how the spatial model looks in terms of aesthetics, but how it facilitates new understandings of, in this case, drum-based improvisation. We outline the process of exploration in meeting this aim below.

2. THE IMPROVSCOPE PROJECT

This research, operates within the modality of 'research *through* design (Downton 2003)' at RMIT University Spatial Information Architecture Laboratory (SIAL) Sound Studios between the authors' creative practice as practicing musician (drummer) and as a practicing architect. The 'ImprovScope' project involves first-person explorations of the author's improvisations on the electronic drum kit. This scoping out of the authors' solo improvisational capacities will serve the dual purpose of contributing to the body of knowledge in relation to musical improvisation and as the basis for further creative research projects, musical compositions and creative works (see www.soundcloud.com/jjham).

Playing drum-based improvisations involves instantaneous recall and action of combinations, patterns and polyrhythms brought about by a bodily engagement in the interface of the drum kit via a pair of drum sticks (hands) and feet. Solo improvisation relies on 'closed' skills (Pressing 1987) which operates without reference to the environment and confounding factors of working with, from and to other musicians. We focus on solo playing for this paper, however acknowledge the limitations of playing alone in the studio. Further project work will extend this research into the drumming of others using the same template, however we focus on the limited context of one drummer for this paper.

King Crimson and Yes drummer, Bill Bruford recently completed his PhD on the creative practice of expert drummers. Bruford creates a cultural psychology of the western kit drummer in order to reveal aspects of creativity in performance (Bruford 2015). This research provides a comprehensive basis for both validating drummers as artists and enabling understandings of the art of kit drumming.

The skills of an experienced drum kit practitioner reflects Donald Schon's concept of 'tacit knowing in action' (Schön 1983), wherein a set of 'referent (Pressing 1987)' patterns and phrases are recalled tacitly (without necessarily knowing one is doing it) and brought into action in response to internal and external stimuli. 'Referent' patterns and phrases (riffs) are the 'go to' repertoire that has been learnt, referenced (copied) from others, adapted, built up, evolved and stylized over the players' career. The quality and quantity of these referents vary greatly between players of different skills and experience. They define the players' style and, more than that: they become the player. For example, the dynamics, signature patterns and phrases of Bill Bruford are completely different to those of Terry Bozzio. Both drummers are known for their unique style, and this style is founded on the recall and application of their referent patterns and phrases within certain musical contexts. The hypothetical question arises as to how both these drummers would respond if asked to play a series of one-minute improvisations at 100 beats per minute? How could these be compared and contrasted and how could their unique styles be identified?

The first author, a musician and architect with approximately 35 years experience playing drums, acted as the drummer for this project. The project involved working on a basic template of playing 100 drum patterns in 4/4 at 100 beats per minute for 100 beats (60 seconds) across three contexts.

1. Beat and Fill: playing a range of improvised beats and fills at 100 BPM for one minute
2. Drum Solo: playing a range of improvised drum solo's at 100 BPM for one minute, and;
3. Studio Beats: playing a range of improvisations to an overlay of three separate improvised guitar tracks at 100 BPM for one minute.

The rationale was to provide some form of template for the improvisations that allowed enough time for a reasonable level of creative expression in drum improvisations, progressing through the variable stages of initiation, ideation, formation, thematic development, repetition wind down and completion. 100 BPM provided a reasonable tempo for playing that could be used across contexts. We acknowledge that different tempi and durations will produce different data sets, however by templating the research fewer variables come into play.

Drum improvisations were played on a Roland TD20™ digital drum kit recorded in Reaper™ DAW on an iMac desktop through a Focusright Pro 40™ audio interface connected by Firewire cable. Drums were recorded in MIDI format, with drum sounds modeled in the Drumasonic plugin in Kontakt™. Improvisations, which took place over a period of months in 2015 were recorded on a Go Pro™ camera and notes and reflections taken afterwards.

The Roland TD20™ drum kit consisted of six drum pads, snare, kick and four cymbal pads (refer figure 1).

Figure 1. Roland TD20 Digital Drum kit layout

2.1 Collation and Tagging

We conducted a series of affordance experiments to test ways of translating music into other domains for processing. The project resulted in the creation of a sample set of 170 one-minute drum improvisations across the three contexts. The first affordance experiment involved txt-based tagging. Drum tracks were listened to and analysed to identify signature 'referent' drum patterns and phrases, then cut and exported as individual MIDI files and tagged according to a schema, of 'timing'; 'style'; 'complexity' and 'beat type' along with other descriptors. MIDI files were then imported into a Devonthink Pro™ database and the Ammonite™ App used to provide tag clouds to allow the easy searching of tags across the range of phrases. (Refer Figure 2, below).

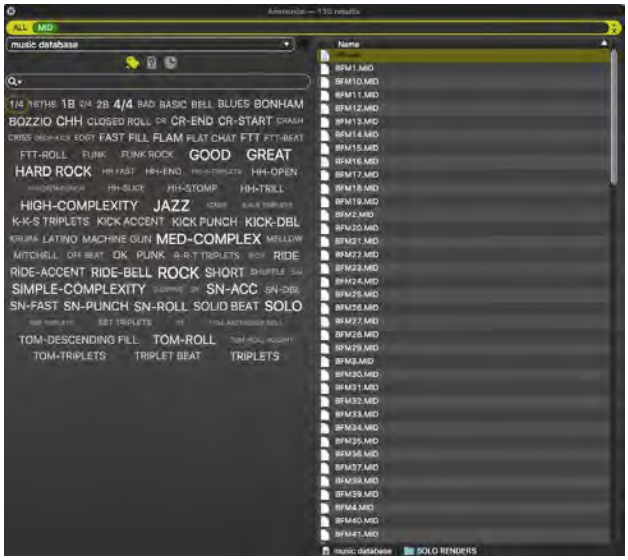


Figure 2. Ammonite Tag cloud (left) and Devonthink Pro file list (right)

These untagged MIDI files in and of their own constitute a series of data readable within a MIDI player or DAW. Importing MIDI files into MuseScore to produce scores enables trained musicians to read the notes, however contains none of the information regarding velocity, microtiming and the form and shape of improvisations. Tagging relies on word as descriptors for patterns and phrases however we needed to push the exploration further into the spatial domain in order to realize our research aims.

3. AFFORDANCE THROUGH PARAMETRIC SPATIAL DESIGN TOOLS

Breithaupt (1987) defined nine strategies for drumset improvisation that provide a context for analysis: Dynamics; Tempo (rate of strokes); Accents; Space; Double strokes/ sticking patterns; Hand to foot distribution; Motion; Special effects and Random use of all elements. These have been used to analyse drumming style of Jeff Porcaro (Artimisi 2011). Drum playing occurs as events in space and time, comprising form, shape, colour and texture. The question is, can spatial design tools offer insights into the morphology of individual drum improvisation, in terms of form, space and clustering of beats, phrases and patterns across large sample sets?

Muecke and Zach (2007) note: '*Performed in real time, music never exists as a whole at any given moment, but rather unfolds in a linear manner over time, and assumes an entity only in retrospect, in the memory of the listener or the performer. However reading a compositional music score is a process closer to perceiving space, as it exists as a whole at any given moment but may be retained by the observer only by a process of observation over time, walking around through, and above it'.* The problem is that many musicians (for example Jimi Hendrix, B.B. King and James Brown) cannot read music scores and that this inability to read may limit opportunities for learning, analysis and reflection-on-action (Schön 1983). In particular, we are interested in the affordance of opportunities for understanding for non-reading visual-dominant musicians.

Each musical interpretation tool, including traditional musical scores is founded on innate limitations and opportunities. For the mathematically inclined the Midi Toolbox (Eerola and Toiviainen 2004) allows Matlab™ analysis of note distributions, melodic contouring, tonality and a range of other functions. The Bol Processor 'produces music with a set of rules (a compositional grammar) or from text 'scores' typed or captured from a MIDI instrument (Bel 1998)'. Matlab™ and the Bol Processor offer mathematical analytical opportunities, however these require interpretive skills that many don't have.

Given the context of this research at the intersection of musical and spatial design domains, it is natural for us as spatial designers to search within their domain for tools, methods and media that provide affordance (Gibson 1979) to the complexities of drum improvisation. An 'affordance is what one system (say, an artifact) provides to another system (say, a user). The concept of affordance is relational because of the complementarity entailed between two interacting systems (Maier and Fadel 2009).'

3D CAD modeling provides affordances to break down the 'invisible perspectival hinge that is always at work between common forms of representation

and the world to which they refer' thus acting to limit comprehension in design processes. Working beyond the limitations of the perspectival hinge requires training and experience. A person's ability to interpret the three-dimensional reality of a building through the representations of plan, section and elevation are at the core of the concept of the 'perspectival hinge'.

We are interested in the proposition of a 'musico-perspectival hinge' within the musical domain that acts between the musical output and the score. Just as architecture students limit their understandings of their building design through orthogonal drawings, musicians may also be limited in their understandings through traditonal scores. As spatial designers, we interface with design information visually and spatially every day- in the form of drawings, models, written notes and in Computer-Aided Design (CAD). We have thus attempted to mediate this 'musico-perspectival hinge' to bring into the spatial domain the outcome of split-second decisions on timing, drum selection and phrasing, complex overlays of polyrhythms and subtle velocity changes on the digital drum kit.

3.1 The Pure Data™ Patch

The second affordance experiment involved the development of a Pure Data™ patch that reads, records and plays MIDI drum improvisations, outputting MIDI data to a display window. This window provides an isometric projection of the MIDI data based on a representation of the digital drum kit with snare at centre, kick at bottom and radiating arrays of tom toms and cymbals (Refer Figure 3). Velocity is represented by note colour. Parameters such as velocity and note diameter can be adjusted. This patch brought drum-based MIDI information into the dynamic visual domain however we found imitations in readability, time fixation and location of events in time. Whilst the patch allowed 3D representations on screen, we found that it did not provide much meaningful information or means of breaching the theoretical musico-perspectival hinge.

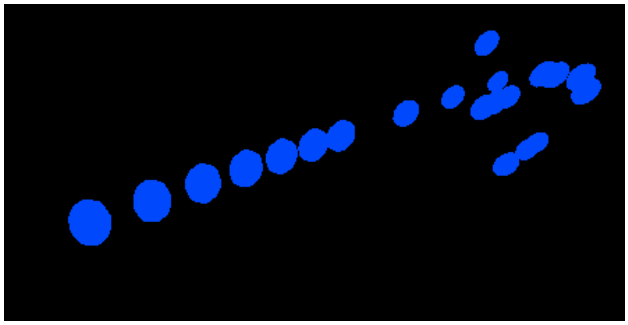


Figure 3. Pure Data Extended patch display

3.2 The 'ImprovSpace' Grasshopper Script

The third affordance experiment involved the use of Rhinoceros3D™ v5 with the Grasshopper™ (GH) plugin to build a flexible parametric framework that enables spatialisation of MIDI drum improvisations in plan, section, elevation, perspective and isometric projections. Parametric digital design, unlike other forms of 3D CAD modeling, (in this case) is based on basic user defined mathematical rules, which can be manipulated to alter 3D virtual objects. These basic user defined mathematical rules are based on the parameters within raw MIDI file reformatted into a comma separated value (.csv) file using the open-source Sekaiju application. The Grasshopper™ script reads data from columns in the .csv file for tempo, 'drum note', 'velocity', 'note on' and 'note off' over time to 1/1000ths of a beat accuracy. These data are separated and sorted using standard Grasshopper components to result in a series of points in space for each drum note with velocity and note length data attached. The virtual sequencing within the redefined midi.csv matrix allowed for basic manipulation of virtual spatial representations of instrumental sounds. The GH interface allows the additive manipulation of spatial data and the ability to use 'sliders' to review multiple .csv files. MIDI data can also be compiled so that several drum improvisations can be overlaid onto each other to identify repeated themes and patterns.

Once the MIDI.csv files are imported into GH as data streams, many spatialization options are available. A process of design exploration examined the ways in which drum improvisations could be represented meaningfully as vectors, curves, surfaces and meshes. A key attribute of parametric modeling is the ability to design spatial elements through the manipulation of the parameters. Thus models are flexible, and can adjust to enable the exploration of options during the design process. The framework we have developed allows this across multiple parameters easily and effectively.

We adopted a stylized representation of the drum kit that related directly to the playing of the drum kit (See figure 1, above). The ImprovSpace GH script stylises this representation of the drum kit in a way that is intended to be easy to interpret and understand as a semiotic or symbolic representation, and in 3D as a representation of patterns and phrases over time. The layout is based on the Snare drum at the centre of two Golden Section spirals- one to track tom toms and a second to track cymbals (Refer Figure 4, below). These drums are colour coded to allow further visualization and understanding of the drums.

Figure 4. Schema for the representation of the Digital drum kit

This spiral frame in the X-Y plane is arrayed into the Z plane, with each array representing a beat or solo at 100 BPM for the bar length of the improvisation, pattern or

phrase. This enables a temporal fixation to the theoretical time structure of the Beat and Fill, Drum Solo, Studio Beat or smaller phrases derived from longer improvisations. Drum improvisations are represented in plan (top left), elevations (bottom left and bottom right) and isometric, perspective or other projection (top right) (Refer Figure 5) and can be animated and 3D printed as haptic music scores.

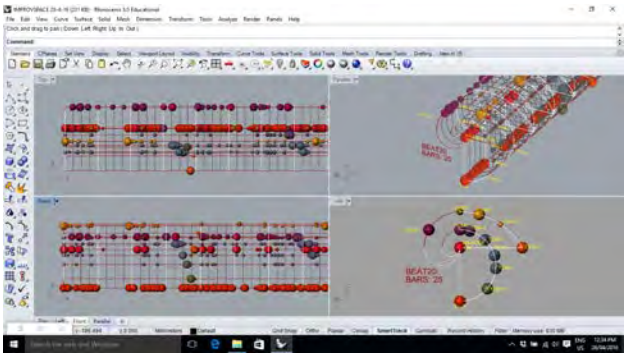


Figure 5. Rhino3D ImprovSpace GH script screenshot of Beat and Fill improvisation No. 20.

100 'referent' samples from the 'Beat and Fill' and 'Solo' improvisational contexts have been extracted to provide a lexicon of the scope of improvisational referent drum patterns and phrases. Tags associated with these samples have been added to a separate .csv file then read into the ImprovSpace GH script. This allows for the cross-referencing of text-based information with the spatialised output. Using the Slider in the GH script, users can cycle through the 200 samples, quickly accessing text tags, plan, elevations and isometric projections (Refer Figure 6). The complementarity of this interface allows shortcomings of understanding in one viewpoint to be compensated for in another.

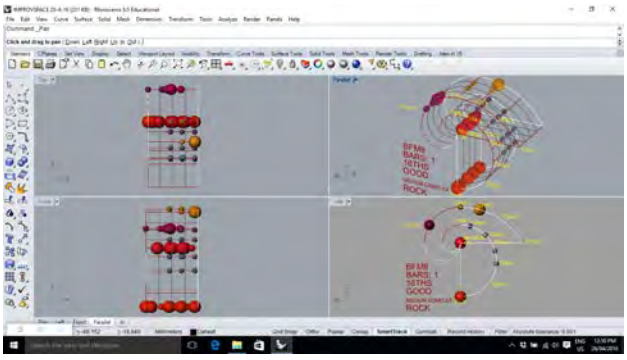


Figure 6. Rhino3D ImprovSpace GH script screenshot of Beat and Fill Sample No. 8.

3.3 Design Development

The development of the ImprovSpace GH script was a design exercise undertaken by musician/ spatial designers. This design process involves reflection on the usability of the script, problems and opportunities for further research. The development of the script brought a Eureka moment when, after working through tagging and the PD patch, a full 3D model of each improvisation was available. As a non-reading spatially-dominant musician, opportunities were created to access multiple 'referent' patterns and phrases quickly and effectively. By bringing these into the spatial dimension, we argue that they have more meaning for us. Patterns can be overlaid onto one another, quantized patterns compared with unique user patterns, and users can be compared with one another.

This can be provided for in the home studio using relatively cost-effective tools such as a PC laptop, Reaper™ DAW, Rhino3D™ and MS Office™ suite. The limitations of computational capacity are felt when large compilations are inputted into the GH script. This is a function of the large number of drum events being played being computed. In the case of comparing all drum Solo's, the computer locks up when computing 120,000 drum events, making analysis difficult.

The ability to rotate views, zoom in, pan in the Rhino3D viewports provides considerable affordance to understanding drum improvisations in the spatial dimension. This is further enhanced by the multiple spatial outputs available and the flexible modeling of notes, velocity and duration. As a framework, this provides considerable research potential for other musicians to develop understandings of their music within a reflective and educational context.

4. FURTHER RESEARCH

This particular method of data manipulation is not limited to just drum beats – the midi.csv format can support other instruments where the pitch of the notes remain the same for their duration ('note on' to 'note off'). More sophisticated clustering of riffs, sequence similarity criterion, classification of certain musical phrases based on predefined user criterion (such as tempo or velocity variation e.g. *accelerando, ritardando, crescendo, decrescendo*) could be applied. However, the computational effort required for such classifications may require lower level computer languages (more computationally efficient than the graphical scripting capabilities in Grasshopper3D). Though, embedding custom components within the Grasshopper environment to manage heavy computational tasks may be an option.

The next stage of the research utilizes the ImprovSpace script to compare the styles of different drummers. We have conducted further affordance experiments in the area of 3D printing and Virtual Reality applications.

As musicians and spatial designers, we are interested in cross-over opportunities into the realm of digital craft. Although the project started on the premise that the focus was on process and not product, as a by-product, we have found a unique means of generating complex spatial forms that reside within contemporary musical practice. As spatial designers, this improvised outcome has triggered unforeseen artistic spatial design opportunities (Refer figure 7, below). These opportunities are being explored in association with musical opportunities derived from composing with the MIDI samples using layering, processing and other techniques in Kontakt Battery. Further potential is being explored in the 3D printing of drum improvisations, patterns and phrases and the capacity for these to be able to be read and interpreted by musicians.

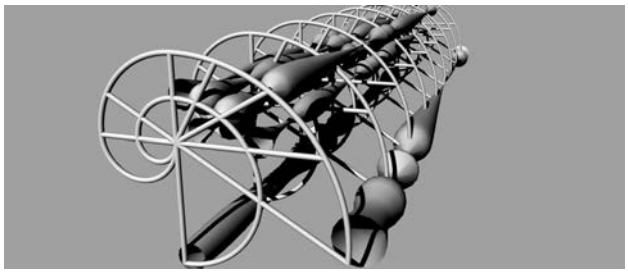


Figure 8. Rendering of lofted artistic representation.

5. CONCLUSIONS

We have outlined a series of creative practice affordance experiments that explore the translation of music into the spatial domain. This has culminated in the development of the ImprovSpace Grasshopper script that utilizes a 3D symbolic representation of the digital drum kit that can be read in plan, section, elevation and 3D. This parametric computational framework allows the flexible manipulation of the parameters of drum-based improvisations to be adjusted to provide affordance to new insights into the elements of micro-timing, polyrhythm, drum selection and other factors that make up an individual style. The framework allows non-reading visually and spatially dominant people opportunities for further understandings of their lexicon of patterns and phrases that make up their style. As such, this appropriation of technologies of the spatial design domain gives meaning to the 'infinite art of improvisation' within a musical domain. Although we focus on the development process for this paper, we see considerable opportunities for this musico-spatial design practice to enable deeper understandings of both domains, and the spaces between. Thus, the research presented forms the basis for further exploration in the form of 3D printing, cross-drummer comparisons and musical and spatial design experimentation.

Acknowledgments

The authors express gratitude for the Australian Government Post-Graduate Scholarship, which provided the funding for this research.

6. REFERENCES

[1] Artimisi, A. B. (2011). "The study of Jeff Porcaro's musical style and the development of an analytical model for the study of drum set style in popular music."

[2] Bel, B. (1998). "Migrating musical concepts: an overview of the bol processor." Computer Music Journal: 56-64.

[3] Bermudez, J. and K. King (2000). "Media interaction and design process: Establishing a knowledge base." Automation in Construction **9**(1): 37-56.

[4] Bruford, W. (2015). Making it Work: Creative music performance and the Western kit drummer. School of Arts, Faculty of Arts and Social Sciences Surrey UK, University of Surrey. **Doctor of Philosophy**.

[5] Downton, P. (2003). Design research, RMIT Publishing.

[6] Eerola, T. and P. Toiviainen (2004). MIR In Matlab: The MIDI Toolbox. ISMIR.

[7] Ferschin, P., et al. (2001). Translating music into architecture. The third International Mathematics and Design Conference M&D2001: digital, hand, eye, ear, mind. Deakin University Geelong, Australia, Mathematics & Design Association.

[8] Fowler, M. (2011). "Appropriating an architectural design tool for musical ends." Digital Creativity **22**(4): 275-287.

[9] Lab-au.com (2015). "The shape of sound." Retrieved 24th April, 2015, from http://lab-au.com/theory/article_soundscapes/.

[10] Levy, A. J. (2003). "Real and Virtual Spaces Generated By Music." International Journal of Architectural Computing **1**(3): 375-391.

[11] McCullough, M. (1998). Abstracting craft: The practiced digital hand, MIT press.

[12] Morgan, M. H. (1960). Vitruvius. The ten books on architecture, Dover New York.

[13] Muecke, M. W. and M. S. Zach (2007). Essays on the Intersection of Music and Architecture, Lulu. com.

[14] Pérez-Gómez, A. and L. Pelletier (1997). Architectural Representation and the Perspective Hinge. Cambridge, Massachusetts & London, The MIT Press.

[15] Pressing, J. (1987). "Improvisation: methods and models." John A. Sloboda (Hg.): Generative processes in music, Oxford: 129-178.

[16] Schön, D. A. (1983). The reflective practitioner: How professionals think in action, Basic books.

DeepGTTM-II: Automatic Generation of Metrical Structure based on Deep Learning Technique

Masatoshi Hamanaka
Kyoto University
hamanaka@kuhp.kyoto-u.ac.jp

Keiji Hirata
Future University Hakodate
hirata@fun.ac.jp

Satoshi Tojo
JAIST
tojo@jaist.ac.jp

ABSTRACT

This paper describes an analyzer that automatically generates the metrical structure of a generative theory of tonal music (GTTM). Although a fully automatic time-span tree analyzer has been developed, musicologists have to correct the errors in the metrical structure. In light of this, we use a deep learning technique for generating the metrical structure of a GTTM. Because we only have 300 pieces of music with the metrical structure analyzed by musicologist, directly learning the relationship between the score and metrical structure is difficult due to the lack of training data. To solve this problem, we propose a multidimensional multitask learning analyzer called deepGTM-II that can learn the relationship between score and metrical structures in the following three steps. First, we conduct unsupervised pre-training of a network using 15,000 pieces in a non-labeled dataset. After pre-training, the network involves supervised fine-tuning by back propagation from output to input layers using a half-labeled dataset, which consists of 15,000 pieces labeled with an automatic analyzer that we previously constructed. Finally, the network involves supervised fine-tuning using a labeled dataset. The experimental results demonstrated that the deepGTTM-II outperformed the previous analyzers for a GTTM in F-measure for generating the metrical structure.

1. INTRODUCTION

We propose an analyzer for automatically generating a metrical structure based on a generative theory of tonal music (GTTM) [1]. A GTTM is composed of four modules, each of which assigns a separate structural description to a listener's understanding of a piece of music. These four modules output a grouping structure, metrical structure, time-span tree, and prolongational tree. As the acquisition of a metrical structure is the second step in the GTTM analysis, an extremely accurate analyzer makes it possible to improve the performance of all later analyzers.

We previously constructed several analyzers that enabled us to acquire a metrical structure such as the automatic time-span tree analyzer (ATTA) [2] and fully automatic time-span tree analyzer (FATTA) [3]. However, the performance of these analyzers was inadequate in that musicologists had to correct the boundaries because of numerous errors.

For this paper, we propose the deepGTTM-II with which we use deep learning [4] to improve the performance of generating a metrical structure from a score. Unsupervised training in the deep learning of deep-layered networks called pre-training aids in supervised training, which is called fine-tuning [5].

Our goal was to develop a GTTM analyzer that enables us to output the results obtained from analysis that are the same as those obtained by musicologists based on deep learning by learning the analysis results obtained by musicologists. We had to consider three issues in constructing such a GTTM analyzer.

- **Multi-task learning**
  A model or network in a simple learning task estimates the label from an input feature vector. However, metrical strength of each beat can be found in every beats. Therefore, we consider a single learning task as estimating whether one beat can be a strong beat or weak beat. Then, a problem in detecting a metrical structure can be solved by using multi-task learning.

  Subsection 4.3 explains multi-task learning by using deep learning.

- **Hierarchical metrical structure**
  A hierarchical metrical structure is generated by iterating the choice of the next level structure. The next level structure is recursively generated using the previous structure. However, when we use learning with a single standard network of deep learning, it is difficult to lean a higher level structure because many network representations are used for learning a lower level structure

  Subsection 4.2 explains how to learn a higher level structure.

- **Large scale training data**
  Large-scale training data are needed to train a deep-layered network. Labels are not needed for pre-training the network. Therefore, we collected 15,000 pieces of music formatted in musicXML from Web pages that were introduced in the MusicXML page of MakeMusic Inc [6]. We needed labeled data to fine-tune the network. Although we had 300 pieces

with labels in the GTTM database [7], this number was too small to enable the network to learn.

Subsection 4.1 explains how we collected the data and how we got the network to learn effectively with a small dataset.

- **GTTM rules**
  A GTTM consists of several rules, and a beat that is applied to many rules tends to be strong in metrical structure analysis. As a result of analysis by musicologists, 300 pieces in the GTTM database were not only labeled with the correct metrical structure but also labeled with applied positions of metrical preference rules. Therefore, the applied positions of metrical preference rules were helpful clues for estimating whether one beat can be strong or weak.

  Subsection 4.2 explains how the network learned with the metrical preference rules.

- **Sequential vs. recurrent models**
  There are two types of models, i.e., recurrent and sequential, that can be used for analyzing a hierarchical metrical structure. The recurrent neural network provides recurrent models, which are suitable for analyzing a metrical structure in which cyclical change results in strong and a weak beats. However, the recurrent neural network is difficult to train and training time is very long. On the other hand, sequential models, such as deep belief networks (DBN), are not suitable for detecting the repetition of strong beats. However, the DBN is very simple and performs well in detecting the local grouping boundary of the GTTM in deepGTTM-I.

  Therefore, we choose the DBN for analyzing the metrical structure of a piece. Subsection 4.2 explains how the DBN is trained for analyzing metrical structure.

The results obtained from an experiment suggest that our multi-dimensional multi-task learning analyzer using deep learning outperforms the previous GTTM analyzers in obtaining the metrical structure.

The paper is organized as follows. Section 2 describes related work and Section 3 explains our analyzer called the deepGTTM-II. Section 4 explains how we evaluated the performance of the deepGTTM-II and Section 5 concludes with a summary and an overview of future work.

2. RELATED WORK

We briefly look back through the history of cognitive music theory. The imprecation realization model (IRM) proposed by Narmour abstracts and expresses music according to symbol sequences from information from a score [8, 9]. Recently, the IRM has been implemented on computers and its chain structures can be obtained from a score [10]. On the other hand, the *Schenkerian* analysis analyzes deeper structures called "Urline" and "Ursatz" from the music surface [11]. Short segments of music can be analyzed through Schenkerian analysis on a computer [12]. There is another approach that constructs a music theory for adopting computer implementation [13, 14].

The main advantage of analysis by a GTTM is that it can acquire tree structures called time-span and prolongation trees. A time-span or prolongation tree provides a summarization of a piece of music, which can be used as the representation of an abstraction, resulting in a music retrieval system [15]. It can also be used for performance rendering [16] and reproducing music [17]. The time-span tree can also be used for melody prediction [18] and melody morphing [19].

The metrical structure analysis in a GTTM is a kind of beat tracking. Current methods based on beat tracking [20–23] can only acquire the hierarchical metrical structure in a measure because they do not take into account larger metrical structures such as two and four measures.

Our ATTA [2] by integrating a grouping structure analyzer and metrical analyzer. The metrical structure analyzer has 18 adjustable parameters. It enables us to control the priority of rules, which enables us to obtain extremely accurate metrical structures. However, we need musical knowledge like that which musicologists have to properly tune the parameters.

Our FATTA [3] does not have to tune parameters because it automatically calculates the stability of structures and optimizes the parameters to stabilize the structures. However, its performance for generating a metrical structure is lower than that of the ATTA.

The σGTTM [24] enables us to automatically detect local grouping boundaries by using a decision tree. The σGTTMII [25] involves clustering steps for learning the decision tree and outperforms the ATTA if we can manually select the best decision tree. The σGTTMIII [26] enables us to automatically analyze time-span trees by learning with a time-span tree of 300 pieces of music from the GTTM database [7] based on probabilistic context-free grammar (PCFG). The pGTTM [27] also uses PCFG, and we used it to attempt unsupervised learning. The main advantages of σGTTMIII and pGTTM are that they can learn the context in difference hierarchies of the structures (e.g., beats are important in the leaves of time-span trees, or chords are important near the roots of the trees.). However, none of these analyzers [7, 24, 25, 27] can generate the metrical structure.

On the other hand, our deepGTTM-I [28] outperforms the ATTA, FATTA, σGTTM, and σGTTMII in detecting local grouping boundaries by introducing deep learning for GTTM analysis. However, it also cannot acquire the hierarchical grouping structure.

In light of this, we introduce a deep learning analyzer for generating the hierarchical metrical structure of a GTTM.

3. GTTM AND ITS IMPLEMENTATION PROBLEMS

Figure 1 Shows a grouping structure, metrical structure, time-span tree, and prolongational tree. The metrical structure describes the rhythmical hierarchy of a piece of music by identifying the position of strong beats at the levels of a quarter note, half note, measure, two measures, four mea-

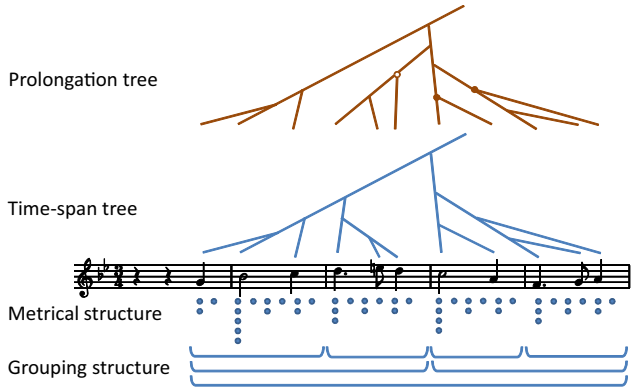


Figure 1. Grouping structure, metrical structure, time-span tree, and prolongation tree

sures, and so on. Strong beats are illustrated as several levels of dots below the music staff.

3.1 Metrical Preference Rules

There are two types of rules in a GTTM, i.e., ”well-formedness” and ”preference”. Well-formedness rules are necessary for the assignment of a structure and restrictions on the structure. When more than one structure satisfies the well-formedness rules, the preference rules indicate the superiority of one structure over another.

There are ten metrical preference rules (**MPRs**): **MPR1** (parallelism), **MPR2** (strong beat early), **MPR3** (event), **MPR4** (stress), **MPR5** (length), **MPR6** (bass), **MPR7** (cadence), **MPR8** (suspension), **MPR9**(time-span interaction), and **MPR10** (binary regularity). **MPR5** has six cases: (a) pitch-event, (b) dynamics, (c) slur, (d) articulation, (e) repeated pitches, and (f) harmony.

3.2 Conflict Between Rules

Because there is no strict order for applying **MPR**s, a conflict between rules often occurs when applying them, which results in ambiguities in analysis.

Figure 2 shows an example of the conflict between **MPRs 5c** and **5a**. The **MPR5c** states that a relatively long slur results in a strong beat, and **MPR5a** states that a relatively long pitch-event results in a strong beat. Because metrical well-formedness rule 3 (**MWFR3**) states that strong beats are spaced either two or three beats apart, a strong beat cannot be perceived at both onsets of the first and second notes.

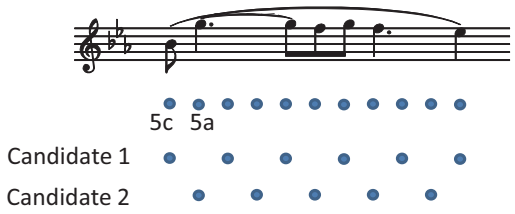


Figure 2. Example of conflict between MPRs

A beat that is applied to many rules tends to be strong in the analysis of a metrical structure. However, the number of rules cannot be determined because the priority of rules differs depending on the context of a piece.

We expect to learn the rule application and priority of rules by inputting a whole song with labels of the applied rules to a deep layered network.

3.3 Ambiguous Rule Definition

Some rules in a GTTM are expressed with ambiguous terms. For example **MPR5** is defined as follows.

> The **MPR5 (Length)**, preference for a metrical structure in which a relatively strong beat occurs at the inception of either
>
> a. a relatively long pitch-event,
> b. a relatively long duration of a dynamic,
> c. a relatively long slur,
> d. a relatively long pattern of articulation,
> e. a relatively long duration of a pitch in the relevant levels of the time-span reduction, or
> f. a relatively long duration of a harmony in the relevant levels of the time-span reduction (harmonic rhythm)

The term ”relatively” in this sense is ambiguous. Another example is that a GTTM has rules for selecting proper structures when discovering similar melodies (called parallelism) but does not define similarity. For example **MPR1** is defined as follows.

> The **MPR1 (Parallelism)**, where two or more groups or parts of groups can be construed as parallel, they preferably receives a parallel metrical structure.

3.4 Context Dependency

To solve the problems discussed in Subsections 3.2 and 3.3, we proposed the machine executable extension of GTTM (exGTTM) and ATTA [2]. Figure 3 is an example of an application of **MPR4**, **5a**, **5b**, and **5c** in the exGTTM and ATTA. By configuring the threshold parameters $T^j(j = 4, 5a, 5b, \ and \ 5c)$, we can control whether each rule is applicable. However, proper values of the parameter depend on the piece of music and on the level of hierarchy in the metrical structure. Therefore, the automatic estimation of proper values of the parameters is difficult.

3.5 Less Precise Explanation of Feedback Link

A GTTM has various feedback links from higher-level structures to lower-level ones. For example **MPR9** is defined as follows.

> The **MPR9 (Time-span Interaction)** has preference for a metrical analysis that minimizes conflict in the time-span reduction.

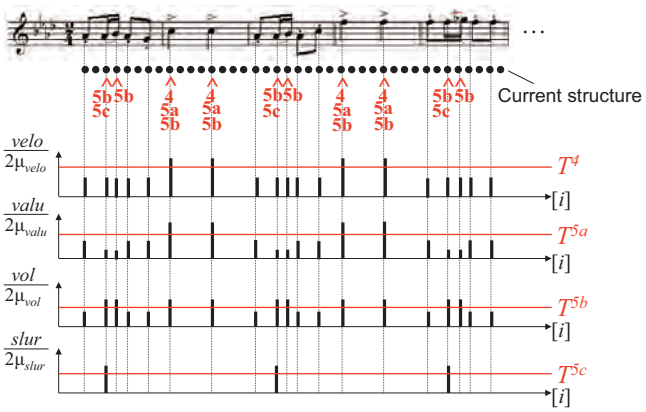


Figure 3. Application of MPR4, 5a, 5b, and 5c in ATTA

However, no detailed description and only a few examples are given. Other feedback links in the GTTM rules are not explicit. For example, analyzing the results of a time-span tree strongly affects the interpretation of chord progression, and various rules are related to chord progression, e.g., **MPR7 (Cadence)** requires a metrical structure in which cadences are metrically stable.

For complete implementation of a GTTM based on deep learning, we have to introduce the feedback link by using recurrent neural network; however, we do not focus on the feedback link in this paper.

4. DEEPGTTM-II: METRICAL STRUCTURE ANALYZER BASED ON DEEP LEARNING

We adopted deep learning to analyze the structure of a GTTM and solve the problems described in Subsections 3.2, 3.3, and 3.4. There are two main advantages in adopting deep learning.

- Learning rule applications
  We constructed a deep-layered network that can output whether each rule is applicable on each level of beat by learning the relationship between the scores and positions of applied MPRs with deep learning.

  Previous analysis systems based on a GTTM were constructed by a researchers and programmers. As described in Subsection 3.3, some rules in a GTTM are very ambiguous and the implementations of these rules might differ depending on the person.

  However, the deepGTTM-II is a learning-based analyzer the quality of which depends on the training data and trained network.

- Learning priority of rules
  Our FATTA does not work well because it only determines the priority of rules from the stability of the structure because the priority of rules depends on the context of a piece of music. The input of the network in the deepGTTM-II, on the other hand, is the score and it learns the priority of the rules as the weight and bias of the network based on the context of the score.

This section describes how we generated a metrical structure by using deep learning.

4.1 Datasets for training

Three types of datasets were used to train the network, i.e., a non-labeled dataset for pre-training, half-labeled dataset, and labeled dataset for fine-tuning (Fig. 4).

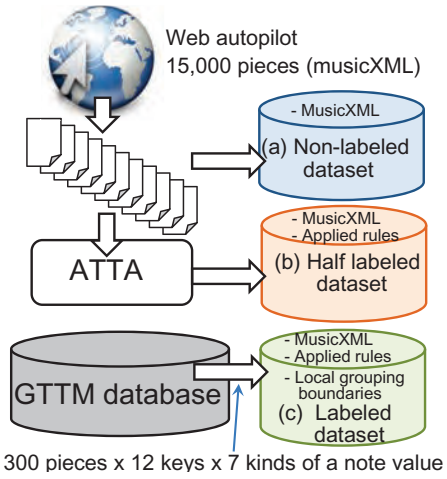


Figure 4. Non-labeled, half-labeled, and labeled datasets

(a) **Non-labeled dataset**. The network in pre-training learned the features of the music. A large-scale dataset with no labels was needed. Therefore, we collected, 15,000 pieces of music formatted in musicXML from Web pages that were introduced on the musicXML page of MakeMusic Inc. [11] (Fig. 4a). The musicXMLs were downloaded in the following three steps.

  (1) Web autopilot script made a list of urls that probably downloaded musicXMLs in five links from the musicXML page of MakeMusic Inc.

  (2) The files in the url list were downloaded after they had been omitted because they were clearly not musicXML.

  (3) All the downloaded files were opened using the script, and files that were not musicXML were deleted.

(b) **Half Labeled Dataset**. The network in fine-tuning learned with the labeled dataset. We had 300 pieces of music with a labeled dataset in the GTTM database, which included musicXML with a metrical structure, and positions to which the MPRs were applied. However, 300 pieces were insufficient for deep learning. Consequently, we constructed a half-labeled dataset. We automatically added the labels of the seven applied rules of **MPR2**, **3**, **4**, **5a**, **5b**, **5c**, and **5d**. These rules can be uniquely applied from a score when we give the threshold values. We used our ATTA to add labels to these rules (Fig. 4b). With the ATTA, the strength of the beat dependent on each MPR can be expressed as

${D_i}^j (j = 2, 3, 4, 5a, 5b, 5c, \ and \ 5d, \ 0 \leq {D_i}^j \leq 1)$. For example, **MPR4** is defined in a GTTM as follows.

> The **MPR4 (Event)**, preference for a metrical structure in which beats of level L_i that are stressed are strong beats of L_i.

We formalized ${D_i}^4$ as follows.

$$ {D_i}^4 = \begin{cases} 1 & velo_i > 2 \times \mu_{velo} \times T^4 \\ 0 & else, \end{cases} \quad (1) $$

where $velo_i$ is the velocity of a note from beat i, μ_{velo} is the average of $velo_i$, and T^j $(0 \leq T^j \leq 1)$ are the threshold parameters to control the those that determines whether the rules are applicable $({D_i}^j = 1)$ or not $({D_i}^j = 0)$. We used 1 as the threshold parameter value $(T^j = 1, \ where \ j = 2, 3, 4, 5a, 5b, 5c, \ and \ 5d)$.

(c) **Labeled dataset**. We collected 300 pieces of 8-bar-long, monophonic, classical music and asked people with expertise in musicology to analyze them manually with faithful regard to the MPRs. These manually produced results were cross-checked by three other experts.

We artificially increased the labeled dataset because 300 pieces of music in the GTTM database were insufficient for training a deep-layered network. First, we transposed the pieces for all 12 keys. We then changed the length of the note values to two times, four times, eight times, half time, quarter time, and eighth time. Thus, the total labeled dataset had 25,200 (= 300x12x7) pieces (Fig. 4c).

4.2 Deep Belief Network

We used a deep belief network (DBN) to generate a metrical structure. Figure 6 outlines the structure for this DBN. The input of the DBN is the onset time, offset time, pitch, and velocity of note sequences from musicXML and grouping structure manually analyzed by musicologists. Each hierarchical level of the grouping structure is separately inputted by a note neighboring the grouping boundary as 1; otherwise, 0.

The output of the DBN formed multi-tasking learning, which had eight outputs in each hierarchical level of the metrical structure, such as seven types of MPRs (**MPR2**, **3**, **4**, **5a**, **5b**, **5c**, and **5d**) and one level of the metrical structure. Individual outputs had two units, e.g., rules that were not applicable (=0) and rules that were applicable (=1), or weak beats (=0) and strong beats (=1).

A metrical structure consists of hierarchical levels, and we added one hidden layer to generate the next structure level. We used logistic regression to connect the final hidden layer (n,n+1,..., n+h) and outputs. All outputs shared the hidden layer from 1 to the final hidden layer. The network was learned in the four steps below. The order of music pieces was changed at every epoch in all steps.

(a) **Pre-training hidden layers from 1 to n.** Unsupervised pre-training was done by stacking restricted Boltzmann machines (RBMs) from the input layer to the hidden layer n. Pre-training was repeated for a hundred epochs using 15,000 pieces in a non-labeled dataset.

(b) **Fine-tuning of MPR 2, 3, 4, 5a, 5b, 5c, and 5d.** After pre-training, the network involved supervised fine-tuning by back propagation from output to input layers. The fine-tuning of MPR2, 3, 4, 5a, 5b, 5c, and 5d were repeated for one hundred epochs using 15,000 pieces in the half-labeled dataset.

(c) **Fine-tuning of one level of metrical structure.** After learning the MPRs, the network involved supervised fine-tuning by back propagation using the labeled dataset of 25,200 pieces at a level of the metrical structure.

(d) **Repeat pre-training and fine-tuning for next level of metrical structure.** If the metrical structure has a next level (more than two dots), add one hidden layer and pre-train the hidden layer using the non-labeled dataset then repeat (b) and (c).

4.3 Multi-dimensional multi-task learning

The DBN we introduced in Subsection 4.2 was a very complex network. The fine-tuning of one level of the metrical structure was multi-task learning. The fine-tuning of each metrical preference rule also involved multi-task learning. Therefore, the fine-tuning of MPRs involved multi-dimensional multi-task learning (Fig. 5).

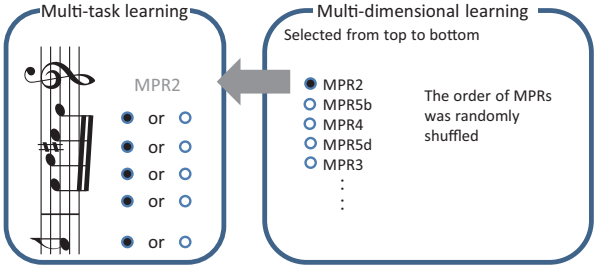


Figure 5. Multi-dimensional multi-task learning

Multi-task learning. The processing flow for the multi-task learning of an MPR or metrical dots involved four steps.

Step 1: The order of the music pieces of training data was randomly shuffled and a piece was selected from top to bottom.

Step 2: The beat position of the selected piece was randomly shuffled and a beat position was selected from top to bottom.

Step 3: Back propagation from output to input was carried out in which the beat position had a strong beat or the rule was applied (=1) or was not (=0).

Step 4: The next beat position or the next piece in steps 2 and 3 was repeated .

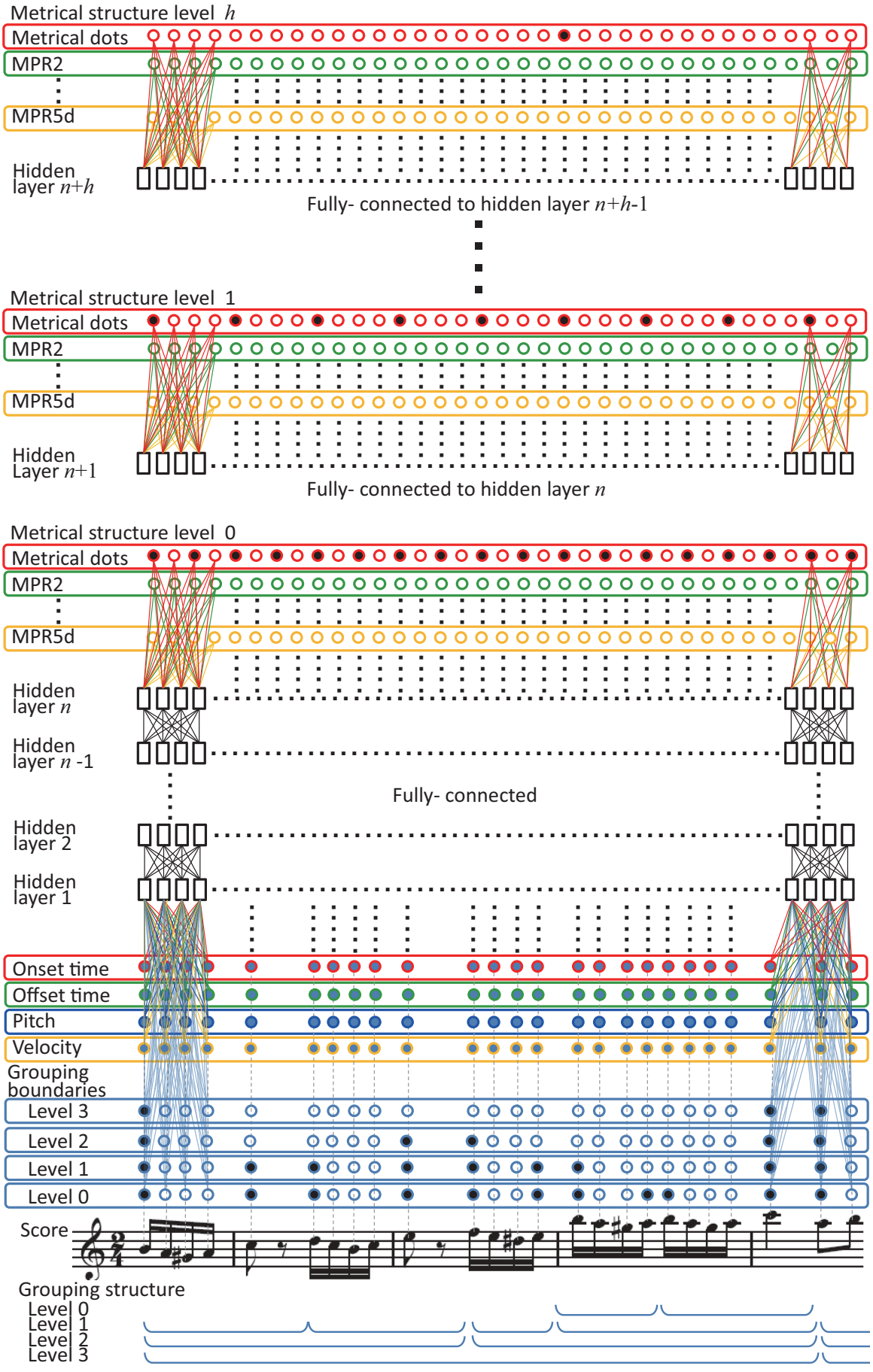


Figure 6. Deep belief network for generating metrical structure

Multidimensional multi-task learning. The processing flow for the multi-dimensional multi-task learning of MPRs involved the following three steps.
Step 1: The order of MPRs was randomly shuffled and a rule was selected from top to bottom.
Step 2: Multi-task learning of the selected MPR was carried out.
Step 3: The next rules in step 1 were repeated.

5. EXPERIMENTAL RESULTS

We evaluated the F_{measure} of the deepGTTM-II by using 100 music pieces from the GTTM database, where the remaining 200 pieces were used to train the network. The F_{measure} is given by the weighted harmonic mean of precision P (proportion of selected dots that are correct) and recall R (proportion of correct dots that were identified).

$$F \quad \text{measure} = 2 \times \frac{P \times R}{P + R} \tag{2}$$

Table 1 summarizes the results for a network that had 11 hidden layers with 3000 units. The ATTA had adjustable parameters and its performance changed depending on the parameters. For the default parameter, we use the middle value of the range of the parameter [2]. The FATTA had no parameters for editing.

The results indicate that the deepGTTM-II outperformed FATTA and ATTA with both default parameters and configured parameters in term of the F_{measure}. This results show that the deepGTTM-II performed extremely robustly.

6. CONCLUSIONS

We developed a metrical structure analyzer called deepGTTM-II that is based on deep learning. The following three points are the main results of this study.

- **Music analyzer based on Deep Learning**
  It has been revealed that deep learning is strong for various tasks. We demonstrated that deep leaning is also strong for music analysis. We will try to implement other music theory based on deep learning. Although we collected 300 pieces of music and analyzed the results of a GTTM by musicologists, the 300 labeled datasets were not sufficient for learning a deep-layered network. We therefore used our previous a GTTM analyzer called ATTA to prepare three types of datasets, non-labeled, half labeled, and labeled, to learn the network .

- **High-accuracy GTTM analyzer without manual editing**
  Previous GTTM analyzers, such as ATTA and σGTTM, require manual editing; otherwise, the performance will be much worse. The F_{measure}s of GTTM analyzers without manual editing, such as FATTA, σGTTM, σGTTMIII, and pGTTM, is too low (under 0.8). On the other hand, the deepGTTM-II shows extremely high performance, which indicates the possibility of its practical use in GTTM applications [15–19, 29]. We plan to implement the entire GTTM analysis process based on deep learning.

- **Multi-dimensional multitask learning**
  We proposed multi-dimensional multi-task learning analyzer that efficiently learns the hierarchical level of the metrical structure and MPRs by sharing the network. Multi-dimensional multi-task learning is expected to be applied to other data that have a hierarchy and time series such as film [30] and discussion [31]. After a network that had 11 layers with 3000 units had been learned, the deepGTTM-II outperformed the previously developed analyzers for obtaining a metrical structure in terms of the F_{measure}.

This work was one step in implementing a GTTM by using deep learning. The remaining steps are to implement time-span reduction analysis and prologational reduction analysis of a GTTM based on deep learning. There are two problems as follows. One is generating tree structures because time-span and prolongation tree structures are more complex than a hierarchical metrical structure. The other problem is the lack of training samples because there are many combinations of tree structures and an unlearned sample sometimes appear in test data. We will attempt to solve these problems and make it possible to construct a complete GTTM system based on deep learning.

In the current stage, we cannot understand the details on why deep learning works extremely well for metrical analysis in a GTTM. Thus, we also plan to analyze a network after a metrical structure is learned.

7. REFERENCES

[1] F. Lerdahl and R. Jackendoff, *A Generative Theory of Tonal Music*, ser. Mit Press series on Cognitive theory and mental representation. MIT Press, 1985.

[2] M. Hamanaka, K. Hirata, and S. Tojo, "Implementing 'a generative theory of tonal music'," *JNMR*, vol. 35, no. 4, pp. 249–277, 2006.

[3] M. Hamanaka, K. Hirata, and S. Tojo, "Fatta: Full automatic time-span tree analyzer," in *Proceedings of ICMC2007*, 2007, pp. 153–156.

[4] G. E. Hinton, S. Osindero, and Y.-W. Teh, "A fast learning algorithm for deep belief nets," *Neural Comput.*, vol. 18, no. 7, pp. 1527–1554, Jul. 2006.

[5] D. Erhan, Y. Bengio, A. Courville, P.-A. Manzagol, P. Vincent, and S. Bengio, "Why does unsupervised pre-training help deep learning?" *JMLR*, vol. 11, pp. 625–660, 2010.

[6] MakeMusic Inc., "Finale," 2016, http://www.finalemusic.com/.

Melodies	deepGTTM-II	ATTA (Default prameters)	ATTA (Configured parameters)	FATTA
1. Grande Valse Brillante	0.94	0.88	0.93	0.88
2. Moments Musicaux	1.00	0.95	1.00	1.00
3. Trukish March	0.98	0.91	0.96	0.96
4. Anitras Tanz	0.90	0.82	0.86	0.82
5 Valse du Petit Chien	0.99	0.87	0.92	0.95
	:	:	:	:
Total (100 melodies)	0.96	0.84	0.90	0.88

Table 1. Performance of deepGTTM-II, ATTA, and FATTA

[7] M. Hamanaka, K. Hirata, and S. Tojo, "Music structural analysis database based on gttm," in *Proceedings of ISMIR2014*, 2014, pp. 325–330.

[8] E. Narmour, *The Analysis and Cognition of Basic Melodic Structures: The Implication-realization Model*. University of Chicago Press, 1990.

[9] E. Narmour, *The Analysis and Cognition of Melodic Complexity: The Implication-Realization Model*. University of Chicago Press, 1992.

[10] S. Yazawa, M. Hamanaka, and T. Utsuro, "Melody generation system based on a theory of melody sequences," in *Proc. of ICAICTA2014*, 2014, pp. 347–352.

[11] S. Heinrich, *Free Composition: New Musical Theories and Fantasies*. Pendragon Pr, 5 2001.

[12] A. Marsden, "Software for schenkerian analysis," in *Proc. of ICMC2011*, 2011, pp. 673–676.

[13] D. Temperley, *The Cognition of Basic Musical Structures*. MIT Press, 2004.

[14] F. Lerdahl, *Tonal Pitch Space*. Oxford University Press, USA, 2001.

[15] K. Hirata and S. Matsuda, "Interactive music summarization based on generative theory of tonal music," *JNMR*, vol. 5, no. 2, pp. 165–177, 2003.

[16] K. Hirata and R. Hiraga, "Ha-hi-hun plays chopin's etude," in *Working Notes of IJCAI-03 Workshop on Methods for Automatic Music Performance and their Applications in a Public Rendering Contest*, 2003, pp. 72–73.

[17] K. Hirata and S. Matsuda, "Annotated music for retrieval, reproduction," in *Proc. of ICMC2004*, 2004, pp. 584–587.

[18] M. Hamanaka, K. Hirata, and S. Tojo, "Melody expectation method based on gttm and tps," in *Proc. of ISMIR2008*, 2008, pp. 107–112.

[19] M. Hamanaka, K. Hirata, and S. Tojo, "Melody morphing method based on gttm," in *Proc. of ICMC2008*, 2008, pp. 155–158.

[20] D. Rosenthal, "Emulation of human rhythm perception," *CMJ*, vol. 16, no. 1, pp. 64–76, 1992.

[21] M. Goto, "An audio-based real-time beat tracking system for music with or without drum-sounds," *JNMR*, vol. 30, no. 2, pp. 159–171, 2001.

[22] S. Dixon, "Automatic extraction of tempo and beat from expressive performance," *JNMR*, vol. 30, no. 1, pp. 39–58, 2001.

[23] M. Davies and S. Bock, "Evaluating the evaluation measures for beat tracking," in *Proc. of ISMIR2014*, 2014, pp. 637–642.

[24] Y. Miura, M. Hamanaka, K. Hirata, and S. Tojo, "Decision tree to detect gttm group boundaries," in *Proc. of ICMC2009*, 2009, pp. 125–128.

[25] K. Kanamori and M. Hamanaka, "Method to detect gttm local grouping boundarys based on clustering and statistical learning," in *Proc. of SMC2014*, 2014, pp. 1193–1197.

[26] M. Hamanaka, K. Hirata, and S. Tojo, "σgttmiii: Learning based time-span tree generator based on pcfg," in *Proc. of CMMR2015*, 2015, pp. 303–317.

[27] E. Nakamura, M. Hamanaka, K. Hirata, and K. Yoshii, "Tree-structured probabilistic model of monophonic written music based on the generative theory of tonal music," in *Proc. of ICASSP2016*, 2016, pp. 276–280.

[28] M. Hamanaka, K. Hirata, and S. Tojo, "deepgttm-i: Local boundaries analyzer based on deep learning technique," in *Proc. of CMMR2016*, 2016, pp. 6–20.

[29] M. Hamanaka, M. Yoshiya, and S. Yoshida, "Constructing music applications for smartphones," in *Proc. of ICMC2011*, 2011, pp. 308–311.

[30] S. Takeuchi and M. Hamanaka, "Structure of the film based on the music theory," in *JSAI2014*, 2014, 1K5-OS-07b-4 (in Japanese).

[31] T. Oshima, M. Hamanaka, K. Hirata, S. Tojo, and K. Nagao, "Development of discussion structure editor for discussion mining based on muisc theory," in *IPSJ SIG DCC*, 2013, 7 pages (in Japanese).

Modulating or 'Transferring' Between Non-octave Microtonal Scales

Todd Harrop
Hochschule für Musik und Theater Hamburg
todd.harrop@hfmt-hamburg.de

ABSTRACT

This paper discusses non–octave-based microtonal scales which can express a septimal minor triad formed by the 6th, 7th and 9th partials of the harmonic series. Three methods are proposed for modulating or transferring between each scale: by pivoting on common tones, building joint chords with pitches unique to each scale, or dynamically changing the sizes of generator and period to transform one scale into another. Motivation for this project was to shed light on two relatively unexplored possibilities in microtonality: scales without octaves, and multiple scales within a single piece of music. With today's computers and synthesizers these areas can be explored more easily. The author borrows a goodness-of-fit strategy for a 6:7:9 chord and chooses three scales that divide the perfect fifth into 8, 13 and 18 equal steps. In addition to septimal triads other common tones are identified, e.g. a major seventh 1/6th-tone sharp, and the paper touches on less obvious manners of modulation. This project may be of interest to composers wishing to explore new facets of microtonality in their work.

1. INTRODUCTION

The conventional scale of twelve equal steps per octave is taken for granted in Western music. It is more economical to build instruments with 12 rather than 19, 31, 43 or 55 keys. Scales of these many notes can approximate the historical temperaments of 1/3rd-, 1/4-, 1/5th and 1/6th-comma meantone, respectively. By slightly shrinking the perfect fifth they offer thirds and sixths that are better in tune than our 12-tone variety.

Prooijen and Carlos, however, showed that by abandoning the octave and shrinking the semitone from 100 to 78 cents or smaller, one can obtain major and minor triads that are closer to being pure [1, p. 51][2]. But what of other triads? And what happens when we lose the octave and its ability to invert or transpose chords by an interval of equivalency? Intervals other than 2/1 are possible yet delicate such as Bohlen–Pierce's 3/1 [3][4] or, theoretically, Moreno's 5/1 and higher [5].

As for contemporary music and theory based around the 3/2 perfect fifth one could look at compositions by Carlos [1] and Serafini [6], or the much older traditional music of Georgia [7, p. 830] and the Eastern Arctic region of Canada [8].

My current work-in-progress, *Apollo,* for percussion and computer, uses three scales that divide the fifth into equal steps but does not presently use the fifth as interval of equivalency for motivic transposition. A previous work, *Bird of Janus,* for Bohlen–Pierce (BP) clarinet, was written in both the BP and Carlos alpha scales [2] and appropriates a common tone of 1170 cents as interval of equivalency. For the new composition strategies other than the use of common tones were desired for moving from one type of tuning to another.

Computer code in `Matlab` and `Max/MSP` assisted me in composing music which could dynamically modulate or 'transfer', to use Darreg's preferred term, between multiple microtonal scales. The impetus came from Wolf's *transitional fifth-squashing* [9] and from *dynamic tuning* by Milne, Sethares and Plamondon [10].

The scales had to satisfy the following criteria: they must (1) be able to express a septimal minor triad with minimal error, (2) have equally-spaced steps (3) which are not 'too small', and (4) not contain an octave. I then considered three strategies for transferring between these scales and presently named them: (1) *communic,* (2) *interharmonic,* and (3) *dynacyclic.* At the risk of sounding whimsical these terms are concise and defined in section 2.2.

2. METHODS

Chords shall be expressed as frequency ratios, e.g. 4:5:6 major triad; dyads or intervals either as ratios, e.g. 5/4 major third and 3/2 perfect fifth; or as intervals from a scale, e.g. 4\12 and 7\12 (note the backslash); scales as either steps per interval, e.g. 12ed2 meaning 12 equal divisions of the 2/1 octave; or as size of scale step in cents, e.g. 100c for the standard semitone with frequency ratio equal to the 12th root of 2.

2.1 Approximating

Prooijen approximated the 4:5:6 major triad with the use of continued fractions to express the answer to equation 1, where f, g and h equal 4, 5 and 6, respectively. Although he found a value of 78.0c Carlos arrived at the same answer a few years later by plotting the results of a goodness-of-fit algorithm using non-integer divisions of the octave, and named this same scale *alpha,* as it is known today.

[1] The title track from her album *Beauty in the Beast* alternates between 9 and 11 equal divisions of the perfect fifth.

[2] 13ed3/1 and 9ed3/2, about 146.3 and 78.0 cents.

$$\frac{h}{f}\log\frac{g}{f} \tag{1}$$

Both Prooijen's and Carlos's methods were applied to a 6:7:9 septimal triad. This triad was chosen because it is the next possible triad in the harmonic series, after 4:5:6, which contains a perfect fifth and which is not playable in the standard 12ed2 scale [11]. This approach yielded three candidate scales and their step sizes were fine-tuned using Benson's formula (eq. 2) for minimizing the mean square deviation from the ideal interval ratios [12, p. 222]. Step size in cents is represented by s while a, b and c represent the number of steps to reach the perfect fifth, septimal major third and septimal minor third:

$$s = 1200 \times \frac{a\log_2\frac{3}{2} + b\log_2\frac{9}{7} + c\log_2\frac{7}{6}}{a^2+b^2+c^2} \tag{2}$$

2.2 Transferring

Pivot tones between scales were found by simply comparing pitch values and noting correspondences when their differences were less than ten cents. This may be called a *communic* strategy, to be used either melodically or harmonically as common notes or chords.

To identify harmonic potential within and between scales `Matlab` was programmed to perform a multi-dimensional search and compare all possible ratios with each scale degree of any given scale. The user must specify a prime limit, [3] a subset thereof, e.g. [2 3 7], or even a set of fractions such as [7/6 3/2]. The user may also stipulate a tolerance for inaccuracy such as 10 cents as well as a maximum *harmonic distance* (eq. 3) [13].

$$HD(f_a, f_b) \propto \log(a) + \log(b) = \log(ab) \tag{3}$$

The output may be used to build a multi-dimensional lattice or Tonnetz for visualizing harmonic structure in the scale (q.v. Johnston, Tenney or Vogel). Such structures could show all scales' pitches, their common pitches, or just their unique pitches.

The latter is the concept behind the *interharmonic* strategy for transferring between scales. To illustrate, imagine scale X to be (0 2 4 6 8 10 12) semitones and scale Y as (0 3 6 9 12)—in other words a whole-tone scale and a diminished seventh chord. Neither scale can play a three-note semitone cluster without assistance from the other, i.e. (2 3 4) or (8 9 10) would require two notes from X and one from Y. The semitone cluster, then, could function as a motivic sonority that signals harmonic transition from one scale to the other. This method is applicable to conventional 12ed2 tuning or between radically different microtonal scales.

For this paper the third strategy will be called *dynacyclic,* however, proper credit is due to Wolf and Milne et al. whose work inspired this project. Wolf in particular was interested in modulating between the Western chromatic scale and an approximation of a Javanese *pelog* scale, as well as in hybridizing them. By casting their differing sizes of fifths as generators and the octave as a period he could morph from one scale to the other by gradually shrinking the fifth from 720 to 700, 685 then 667 cents. This has the effect of pitches splitting apart from a 5ed2 (anhemitonic pentatonic) scale, fanning outward to form a 12ed2 scale, folding inward and colliding into a 7ed2 (anhemitonic heptatonic) scale, expanding out again into 9ed2 and so on.

In section 3.2.3 this technique was adopted as a method for transitioning between non-octave microtonal scales. Coincidentally Wolf was also interested in octave-based scales generated by sequences of tempered 7/6 thirds instead of tempered 3/2 fifths, whereas this paper is focused on fifth-based scales generated by sequences of tempered 7/6 or 9/7 thirds, as will be explained next.

[3] A prime limit is the set of all prime numbers up to a given number, e.g. prime limit 7 is [2 3 5 7].

3. FINDINGS

3.1 Approximating 6:7:9

3.1.1 Continued fractions

Letting f, g and h be 6:7:9 in equation 1 resulted in 0.38018 which, when represented as a continued fraction in abbreviated notation, is [0; 2, 1, 1, 1, 2, 2, ...]. The first four terms were rejected since they indicate 'scales' with a large margin of error. The next three terms rationalized to the convergents 3\8, 8\21 and 19\50, [4] suggesting scales which divide the 3/2 perfect fifth into 8, 21 and 50 equal steps, with the 7/6 ratio occurring on the third, eighth and nineteenth intervals of each.

Only the 8ed3/2 scale was kept, however, since 21ed3/2 is simply the 1/6$^{\text{th}}$-tone scale, i.e. one which contains an octave, and a 50ed3/2 scale would have had a step size of 14 cents which is too small for this project.

The semi-convergents [0; 2, 1, 1, 1, 1] and [0; 2, 1, 1, 1, 2, 1] resulted in 5\13 and 11\29. The 13ed3/2 scale was kept and the 29ed3/2 scale was considered but then rejected for its small step size of 24 cents.

3.1.2 Goodness-of-fit

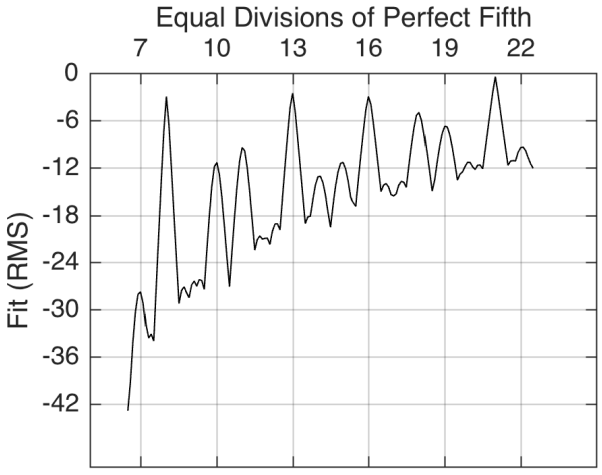


Figure 1. Peaks indicate scales which approximate the ratios 7/6, 9/7 and 3/2.

[4] Read as "3 of 8 (divisions)" etc.

A variation on Carlos's plot of error across step size was used. On the x-axis are non-integer divisions of the 3/2 perfect fifth instead of 2/1 octave, and the y-axis shows accuracy by root-mean-square instead of sum-of-squares. Figure 4 shows strong peaks at 8, 13, 16, 18 and 21ed3/2. In addition to those divisions already found in section 3.1.1 the plot indicates 16ed3/2 and perhaps 18ed3/2. The former is merely a doubling of 8ed3/2 and the latter, though not as accurate, is 'not too bad' and was therefore retained.

In summary three non-octave scales which could approximate a 6:7:9 septimal minor triad were chosen: 8, 13, and 18 equal divisions of 3/2. Finding the interval for 7/6 in the last scale was easily done with equation 4.

$$\frac{18}{\log \frac{3}{2}} \times \log \frac{7}{6} \approx 7 \tag{4}$$

3.1.3 Minimizing deviation

Finally, using Benson's formula (eq. 2) the fifths of each scale were tweaked about one cent in order to minimize inaccuracies for all three intervals in a 6:7:9 triad. For the 8, 13 and 18ed3/2 scales the final step sizes became 87.670, 54.033 and 39.047 cents, respectively.

3.2 Transferring or modulating

3.2.1 Communic

Common tones between scales are the septimal thirds and perfect fifth. Although the 3/2 fifths were perceptually identical the septimal thirds did not fare as well. The most noticeable bump might be heard between 8ed3/2 and 18ed3/2 where the 9/7 major third differs by 8.8 cents or where the 7/6 minor third differs by 10.3 cents.

When comparing scales up to their intervals nearest the octave another common tone of around 1136c was discovered—a major seventh 1/6th-tone sharp. Other affinities are shown in figure 2.

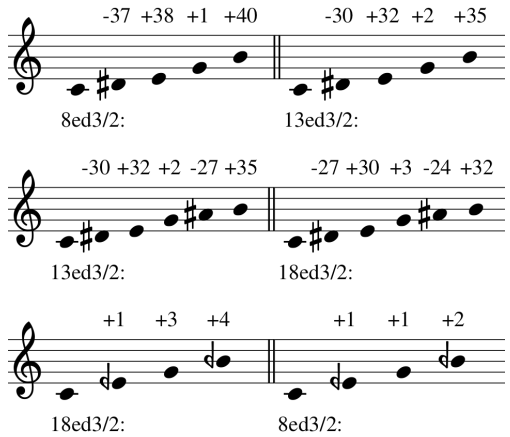


Figure 2. Common pitches between scales. Numerals indicate cents' deviation from standard notation.

Figure 2 shows common pitches which differ by seven cents or less between scales, from 8 to 13 to 18 divisions per fifth, then back to 8. The root, C4, is consistent and the perfect fifth, G4, hardly wavers at all. The 7/6 interval in between is shown as D♯4 about a 1/3rd-tone flat but was too jarring between 18 and 8ed3/2, in the last system, to be considered as a common tone. Surprisingly a very smooth transition between 8 and 18 ed3/2 is possible using a 'neutral third–neutral seventh' chord, as seen in the bottom system.

Additional pivot tones would be available by tolerating more cents' difference or by looking for correlations beyond the quasi-octaves presented.

3.2.2 Interharmonic

The idea was to find not only tones that were common between scales but harmonic structures that could only be possible across simultaneous scale systems. This is complicated but manageable with computation. A cursory search for chords between the 8 and 13 ed3/2 scales reveals two near-perfect versions of the most familiar chords: a just intonation version of an equal-tempered minor triad, 16:19:24, and a nearly perfect equal-tempered major triad. The latter can also be extended to a major seventh chord. Figure 3 shows how the scale pitches complement one another to build these sonorities which are not possible in either scale alone.

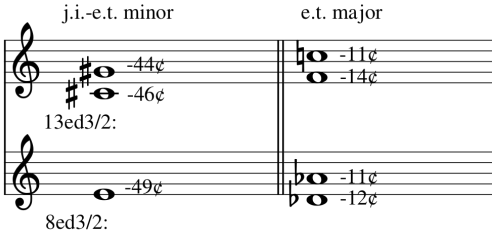


Figure 3. Unique pitches between scales which form chords. Numerals indicate cents' deviation from standard notation.

The top staff is in 13ed3/2 tuning and the bottom in 8ed3/2, and the 'tonic' of both keys is C4. The first measure is a minor triad where the root and fifth come from the 1st and 14th intervals of 13ed3/2, and the third comes from the 4th interval of 8ed3/2. The second measure is a major seventh chord where the root and fifth come from the 1st and 9th intervals of 8ed3/2, and the third and seventh come from the 9th and 22nd intervals of 13ed3/2. These two chords are exceptionally accurate and by loosening the tolerance for cents' deviation many more joint-chords can be constructed.

3.2.3 Dynacyclic

Figure 4 attempts to show what happens when scales are created using a generator of 9/7 and a period of 3/2, although with the tweaked step sizes listed in section 3.1.3. From left to right the scales change over time as the generator is increased from 429.5 to 432.3 then 438.4 cents, and the period is slightly decreased from 702.8 to 702.4 then 701.4 cents. Although these changes might seem minute the differences add up after subsequent generations.

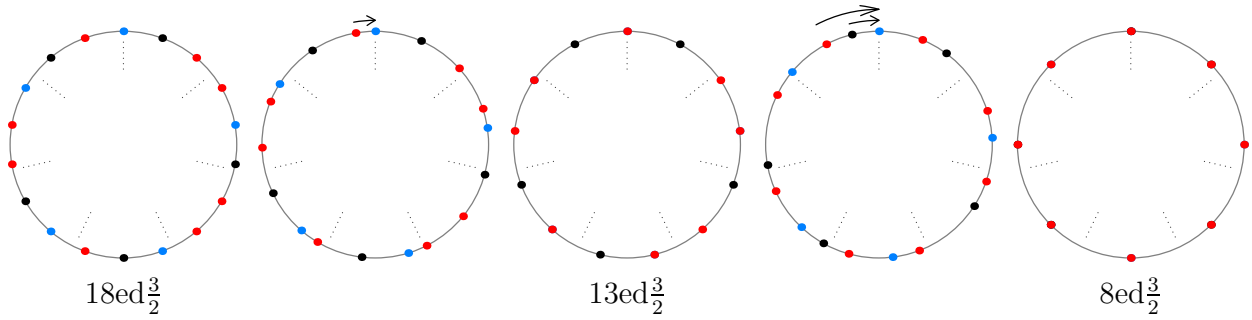


Figure 4. Each circle represents a perfect fifth, where dots are notes from each scale, coloured for ease of locating in the figure. Dotted in-ticks indicate standard semitones for reference, i.e. seven ticks per fifth. Arrows show narrowing intervals which vanish in the next scale (and can be imagined on the other short arc lengths around each circle).

When plotted around a ring, generations 13 to 17 move more quickly than 8 to 12. This can be seen in the figure as the black dots cover the blue dots, and the red dots cover both the black and blue dots. Table 1 summarizes how the 18ed scale twists into a 13ed scale. One can see that five pitches have folded onto five others to form unison pairs:

Ratio	Interval	Size	Generation(s)
1/1	0		0, 13
	1	54c	5
	2		10
	3		2, 15
	4		7
7/6	5	270c	12
	6		4, 17
	7		9
9/7	8	432c	1, 14
	9		6
	10		11
	11		3, 16
	12		8

Table 1. 13ed3/2 scale: intervals (left) and generations of 9/7 (right).

When the process is continued, as from 13ed to 8ed, five more notes fold onto unisons resulting in six pairs and two triplets of 9/7 generations.

Ratio	Interval	Size	Generation(s)
1/1	0		0, 8, 16
	1	88c	5, 13
	2		2, 10
7/6	3	263c	7, 15
	4		4, 12
9/7	5	438c	1, 9, 17
	6		6, 14
	7		3, 11

Table 2. 8ed3/2 scale: intervals (left) and generations of 9/7 (right).

4. DISCUSSION

Microtonal music and research typically assumes that the octave remains in place as interval of equivalency. Had I wanted to consider octave-based scales, instead of non-octave scales, which can express a septimal triad then the following would have been good choices (in order from okay to best): 27, 39, 31, 22, 41, and 36 equal divisions of the octave. The last three are especially good but the sixth-tone scale (36ed2) is practically perfect.

Non-octave scales offer relatively unexplored harmonic territory. Indeed two of the three choices discussed in this paper are already well documented: 8ed3/2 is essentially Morrison's *88c* scale, and 18ed3/2 is better known as Carlos's *alpha prime.*

4.1 Comparison to other scales

The Scala archive is a compendium of over four thousand scales, each comprising a short description and a list of intervals in cents and or ratios. Many are historical temperaments or interpretations of cultural scales, and many others are artistic or academic expressions of tuning concepts.

For example, the archive contains eight scales in the 8ed3/2 family from between 1998–2011. Morrison himself includes scales of 88 cents or slightly less, e.g. least-squares and minimax solutions for approximating intervals 11/9, 10/7 and 7/4 in addition to those intervals in the 6:7:9 triad. With the exception of McLaren's '38th root of 7' they are all excellent choices for expressing the 6:7:9 triad.

There is a half dozen which would belong to the family of what I am calling 13ed3/2, from 1996 to 2007. The thirteenth root of 3/2 or my adjustment is clearly best, and the next closest might be the '62nd root of 7' or '35th root of 3' scale.

Finally there are also eight scales within the 18ed3/2 family of around 39 cent steps from 1996–2007, however the next best option after alpha prime for approximating the 6:7:9 triad would be the '15th root of 7/5' scale, though not a close choice. [5]

The septimal triad could be extended to include the major seventh seen earlier, interpreted as either 27/14 in 7-limit harmony or as 25/13 in 13-limit. This would of course

[5] http://www.huygens-fokker.org/docs/scalesdir.txt, accessed 28 April 2016.

influence the ideal size of scale step and quite likely alter the details of all aspects so far examined.

5. CONCLUSIONS

These scales promise interesting harmonies in and of themselves but modulating or moving in between seems to be a worthy challenge. In the words of Darreg: "Suppose one deliberately composes something which changes tuning-systems in the middle or at several places. It would seem inadvisable to extend the meaning of the already-overburdened word 'modulation' to this novel and startling effect. Hence, *transfer.*" [14]

Leveraging pitches that are common to two or more scales is an obvious and effective method for transferring from one scale or tuning to another. The opposite approach, of using pitches distinct to each scale for building joint sonorities, is another avenue worth pursuing. The example introduced in fig. 3 is admittedly small and could be developed further. Finally the technique of dynamic tuning is getting attention, especially as it naturally goes hand in hand with dynamic timbre matching as explained by Sethares, Milne et al.; of adjusting the overtone structure of a (synthesized) instrument in real-time to "mediate sensory dissonance" of any given scale's interval properties [15][16], or by stretching the timbre such that the partials are no longer integer multiples of a fundamental. This certainly makes sense with alpha prime which could easily pass as a 31-tone scale having a stretched octave of 1210.5 cents.

This paper presented examples of non-octave scales that could express a septimal minor triad but of course the techniques can be applied to other harmonies. And although there seems to have already been a push in the early 1990s for microtonal research, judging by the literature, it is this author's hope that the ideas presented here may inspire composers to reconsider microtonality, especially with the ability of today's technology to smoothly transition from any scale system to another, with or without octaves.

Acknowledgments

The author is grateful for the financial support of the Claussen-Simon-Stiftung.

6. REFERENCES

[1] K. v. Prooijen, "A theory of equal-tempered scales," *Interface*, vol. 7, pp. 45–56, 1978.

[2] W. Carlos, "Tuning: At the crossroads," *Computer Music Journal*, vol. 11, no. 1, pp. 29–43, 1987.

[3] H. Bohlen, "13 tonstufen in der duodezime," *Acustica*, vol. 39, no. 2, pp. 76–86, 1978.

[4] M. Mathews, L. A. Roberts, and J. R. Pierce, "Four new scales based on nonsuccessive-integer-ratio chords," *Journal of the Acoustical Society of America*, vol. 75, p. S10, 1984.

[5] E. Moreno, *Expanded Tunings in Contemporary Music: Theoretical Innovations and Practical Applications*, ser. Studies in the History and Interpretation of Music. Lewiston; Queenston; Lampeter: The Edwin Mellen Press, 1992, vol. 30.

[6] C. Serafini. (2002–16) Carlo Serafini: Futurist composer. [Online]. Available: http://www.seraph.it

[7] J. Jordania, *The Garland Encyclopedia of World Music*. New York and London: Garland Publishing, Inc., 2000, vol. 8 Europe, ch. Georgia, pp. 826–849.

[8] N. Beaudry, *The Garland Encyclopedia of World Music*. New York and London: Garland Publishing, Inc., 2001, vol. 3 The United States and Canada, ch. Arctic Canada and Alaska, pp. 374–382.

[9] D. Wolf, "Why ratios are a good/bad model of intonation," in *The Ratio book: a documentation of The Ratio Symposium, Royal Conservatory, The Hague, 14–16 December 1992*, ser. Feedback Papers, C. Barlow, Ed., vol. 43. Cologne: Royal Conservatory The Hague, 1999, pp. 160–175.

[10] A. Milne, W. Sethares, and J. Plamondon, "Isomorphic controllers and dynamic tuning: Invariant fingering over a tuning continuum," *Computer Music Journal*, vol. 31, no. 4, pp. 15–32, Dec. 2007. [Online]. Available: http://dx.doi.org/10.1162/comj.2007.31.4.15

[11] A. Honingh, "Measures of consonance in a goodness-of-fit model for equal-tempered scales," In Proc. International Computer Music Conference, Tech. Rep., 2003.

[12] D. Benson, *Music: A Mathematical Offering*. Cambridge University Press, 2008.

[13] J. Tenney, *Soundings 13: The Music of James Tenney*. Frog Peak, 1984, ch. John Cage and the Theory of Harmony.

[14] I. Darreg. (1990) Transfer. [Online]. Available: http://www.tonalsoft.com/sonic-arts/darreg/transfer.htm

[15] A. J. Milne, M. Carlé, W. A. Sethares, T. Noll, and S. Holland, *Scratching the Scale Labyrinth*. Springer, 2011.

[16] W. A. Sethares, *Tuning, Timbre, Spectrum, Scale*, 2nd ed. London: Springer, 2005. [Online]. Available: http://www.springer.com/engineering/book/978-1-85233-797-1

SYNCHRONIZATION IN CHAINS OF VAN DER POL OSCILLATORS

Andreas Henrici
ZHAW School of Engineering
Technikumstrasse 9
CH-8401 Winterthur, Switzerland
andreas.henrici@zhaw.ch

Martin Neukom
ZHdK ICST
Toni-Areal, Pfingstweidstrasse 96
CH-8031 Zürich, Switzerland
martin.neukom@zhdk.ch

ABSTRACT

In this paper we describe some phenomena arising in the dynamics of a chain of coupled van der Pol oscillators, mainly the synchronisation of the frequencies of these oscillators, and provide some applications of these phenomena in sound synthesis.

1. INTRODUCTION

If several distinct natural or artificial systems interact with each other, there is a tendency that these systems adjust to each other in some sense, i.e. that they synchronize their behavior. Put more precisely, by synchronization we mean (following [1]) the *adjustment of the rhythms of oscillating objects due to their mutual interaction*. Synchronization can occur in model systems such as a chain of coupled van der Pol oscillators but also in more complex physical, biological or social systems such as the coordination of clapping of an audience [2]. Historically, synchronization was first described by Huygens (1629-1695) on pendulum clocks [3]. In modern times, major advances were made by van der Pol [4] and Appleton [5]. Physically, we basically distinguish between synchronization by external excitation, mutual synchronization of two interacting systems and synchronization phenomena in chains or topologically more complex networks of oscillating objects. Whereas in [6] we discussed the case of two interacting systems, in this paper, we will focus on the case of a (one-dimensional) chain of oscillators with diffusive nearest-neighbor coupling. For a modern overview of the topic of synchronization, see e.g. [7, 8]; an intuitively well accessible example is the synchronization of metronomes [9, 10].

The synchronizability of such a chain of oscillators depends on several parameters of the system, mainly the detuning between the individual frequencies and the strength of the coupling between the oscillators. It can be observed that before completely synchronizing, the chain forms clusters of neighboring masses with similar frequencies. For growing interaction strength, the size of these clusters increases, whereas their number decreases, before at a certain threshold, the whole chain forms a single cluster, which corresponds to the state of complete synchronization.

In addition, we consider chains of oscillators where the coupling does not happen instantaneously, but with a delay. Depending on the value of the delay, significant changes in the dynamics of the system can be observed.

Self-sustained oscillators can be used in sound synthesis to produce interesting sounds and sound evolutions in different time scales. A single van der Pol oscillator, depending on only one parameter (μ, see (1)), produces a more or less rich spectrum, two coupled oscillators can synchronize after a while or produce beats depending on their frequency mismatch and strength of coupling [11,12]. In chains or networks of coupled oscillators in addition different regions can synchronize (the clusters mentioned above), which takes even more time. If the coupling is not immediate but after a delay it can take a long time for the whole system to come to a steady or periodic changing state. In addition all these effects can not only be used to produce sound but also to generate mutually dependent parameters of any sound synthesis technique. In a series of studies (Studien 21.1-21.9) one of the authors (Neukom) investigated these effects.

This paper gives an introduction to the synchronization of such oscillator chains [1, 13], using the van der Pol system as primary example. However, similar effects can be observed for other systems.

2. CHAINS OF COUPLED VAN DER POL OSCILLATORS

2.1 Van der Pol oscillators

Self-sustained oscillators are a model of natural or technical oscillating objects which are active systems, i.e. which contain an inner energy source. The form of oscillation does not depend on external inputs; mathematically, this corresponds to the system being described by an autonomous (i.e. not explicitly time-dependent) dynamical system. Under perturbations, such an oscillator typically returns to the original amplitude, but a phase shift can remain even under weak external forces. Typical examples of self-sustained oscillators are the van der Pol oscillator

$$\begin{aligned} \dot{x} &= y \\ \dot{y} &= -\omega_0^2 x + \mu(1+\gamma-x^2)y \end{aligned} \tag{1}$$

and the Rössler and Lorenz oscillators. Note that in the van der Pol oscillator (1), the parameters μ and γ measure the strength of the nonlinearity; in particular, for $\mu = 0$

we obtain the standard harmonic oscillator. In the case of a single oscillator we usually set $\gamma = 0$, whereas in the case of several oscillators we can use distinct values of γ to describe the amplitude mismatch of the various oscillators. Assuming $\gamma = 0$, in the nonlinear case $\mu \neq 0$, the term $\mu(1 - x^2)y$ means that for $|x| > 1$ and $|x| < 1$ there is negative or positive damping, respectively.

We will discuss an implementation of the van der Pol model (1) in section 4.

2.2 Oscillator Chains

If one considers an entire chain of oscillators, the model equations are for any $1 \leq j \leq n$

$$\ddot{x}_j + \omega_j^2 x_j = 2\mu(p - x_j^2)\dot{x}_j + 2\mu d(\dot{x}_{j-1} - 2\dot{x}_j + \dot{x}_{j+1}) \quad (2)$$

together with the (free end) boundary conditions

$$x_0(t) \equiv x_1(t), \quad x_{n+1}(t) \equiv x_n(t).$$

Sometimes we also use periodic boundary conditions, i.e.

$$x_0(t) \equiv x_n(t), \quad x_1(t) \equiv x_{n+1}(t).$$

Note that in the oscillator chain (2), the parameters d measures the strength of the coupling between neighboring oscillators; in particular, for $d = 0$ we obtain the single oscillator (1), with $p = 1 + \gamma$. As we will see in the following section (see e.g. Figure 3), with growing values of d the oscillators of the chain more likely synchronize.

We assumed different models for the distribution of the frequencies ω_j of the n oscillators, in particular the following ones:

- Linear distribution:

$$\omega_k = \omega_1 + (k - 1)\Delta, \quad (3)$$

  depending on the detuning parameter Δ

- Exponential distribution:

$$\omega_k = \omega_1 + (1 + \Delta)^k, \quad (4)$$

  also depending on the detuning parameter Δ

- Random distribution:

$$\omega_k \sim \mathrm{U}(\omega_{\mathrm{min}}, \omega_{\mathrm{max}}), \quad (5)$$

  i.e. the ω_k's are independent random variables all being distributed uniformly in a certain interval.

We first discuss the linear case (3). In this case, an analytic discussion can be carried out in the case of weak coupling, i.e. $d << 1$ in (2). On can show (see e.g. [13]) that in this case, the second order system (2) can be rewritten as the first order system

$$\dot{z}_j = i\Delta_j z_j + (p - |z_j|^2)z_j + d(z_{j+1} - 2z_j + z_{j-1}) \quad (6)$$

for complex variables z_j, which turns out to be a spatial discretization of the Ginzburg-Landau equation. By writing $z_j = \rho_j e^{i\theta_j}$, the complex system (6) can then be rewritten as a real system for the amplitudes ρ_j and phases θ_j. Global synchronization of the original dynamical system (2) then means that this new system for the variables ρ_j, θ_j has a stable steady state. One shows that this occurs if the condition

$$\left| \frac{\Delta n^2}{8d} \right| < 1 \quad (7)$$

is satisfied, or put differently, that the coupling parameter d has to be above the threshold $\frac{\Delta n^2}{8}$ for the system (2) to be completely synchronized.

2.3 Numerical Simulations

In Figure 1, where the n trajectories are plotted on a $t - x_j-$diagram, one observes the complete synchronization of a chain of $N = 20$ oscillators with linearly distributed frequencies, i.e. according to (3), with $\Delta = 0.002$, and with the coupling $d = 0.5$. The synchronization condition (7) is clearly satisfied. The initial conditions were chosen randomly in the interval $[-1, 1]$. All figures in this section were produced with the pre-implemented methods `ode45`, `ode23s`-and `dde23`-methods of `Matlab`, and in all simulations, we used the values $\omega_1 = 1$, $p = 0.5$, and $\mu = 1$ in (2) and (3).

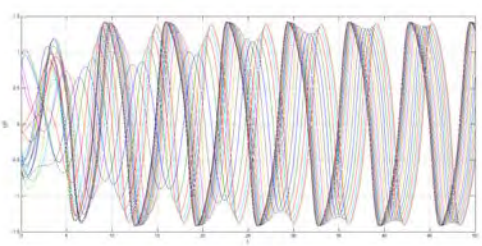


Figure 1. Complete synchronization of the van der Pol chain (2) for $n = 20$, $\Delta = 0.002$, and $d = 0.5$

In Figure 2, where the n trajectories are plotted as colors in a $j - t-$diagram, we consider the case of a chain of $n = 100$ oscillators with linear frequency distribution (3) for the detuning $\Delta = 0.002$ and the coupling $d = 2.5$. The condition (7) is clearly not satisfied, and one can observe the increasing oscillation frequencies for growing j. One also sees the existence of synchonization clusters, i.e. regions of indices j where the respective oscillators have the same frequency.

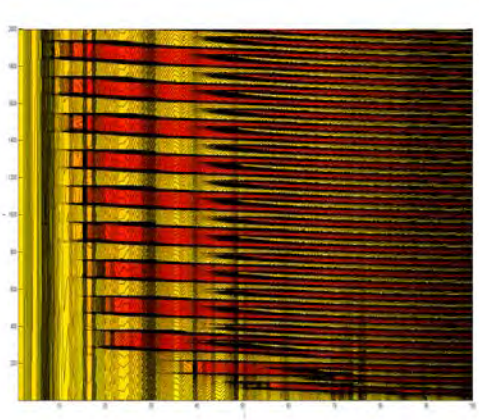


Figure 2. Space-time plot of the chain (2) for $\Delta = 0.002$ and $d = 2.5$

The transition from the cluster to the synchronization regime can be well seen in the following figure, where for different values of the coupling d the oscillator frequencies are plotted vs. the index j:

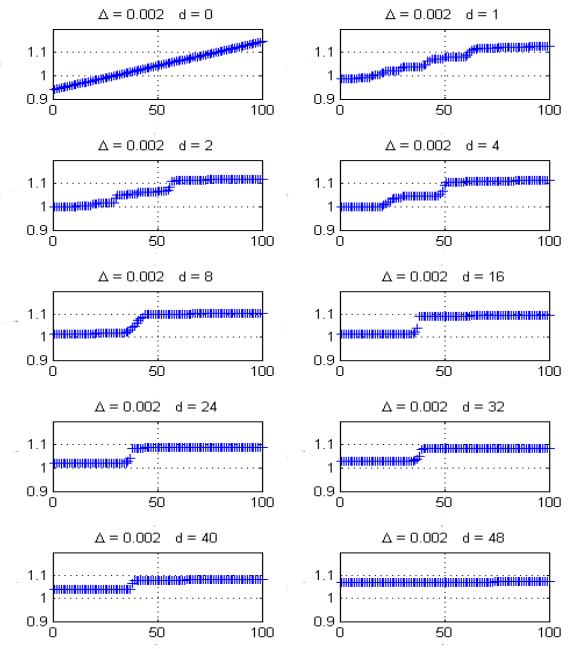


Figure 3. Transition from the cluster formation to the synchronization regime for the chain (2): Space-frequency- (i.e. j-ω_j-) plots for $\Delta = 0.002$ and various values of d

Yet another way to describe the process of synchronization for growing coupling strength is by the *synchronization tree* which plots the coupling strengh vs. the frequencies:

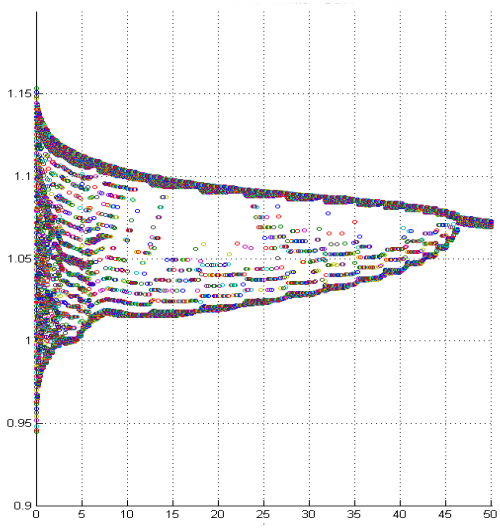


Figure 4. Synchronization tree for the chain (2): Coupling-frequencies- (i.e. d-ω_j-) plots, $1 \leq j \leq n$, with $\Delta = 0.002$

2.4 Influence of delays

If a delay τ is introduced into the system (2), we obtain the system of DDE's (delay differential equations)

$$\begin{aligned}\ddot{x}_j(t) + \omega_j^2 x_j(t) = 2\mu(p - x_j^2(t))\dot{x}_j(t) \\ +2\mu d(\dot{x}_{j-1}(t-\tau_B) \\ -2\dot{x}_j(t-\tau_0) + \dot{x}_{j+1}(t-\tau_A)),\end{aligned} \quad (8)$$

where τ_0, τ_A, and τ_B denote the delays from oscillators $j \to j$, $j+1 \to j$ (backward), and $j-1 \to j$ (forward). Note that we only consider the simplest situation of the delays being constant. We chose $\tau_0 = \tau_A = \tau_B =: \tau$ for our numerical experiments and $\tau_0 = 0$ (no self-delay) and $\tau_A = \tau_B =: \tau$ for our experiments in Max (see section 3). For a study of a single van der Pol oscillator with delayed self-feedback see [14], and a pair of such oscillators has been investigated in the case without detuning [15]. The assumption of a delayed signal transmission or feeedback process is a very natural assumption in certain physical and biological systems, sind theses transmission and feedback processes in general do not occur instantaneously. For an overview over the dynamics of time-delay systems such as (8), see e.g. [16, 17].

Our experiments show that already small values of τ can completely alter pictures such as Figure 2, see Figures 5, 6, and 7 for the cases $\tau = 0.08$, $\tau = 0.09$ and $\tau = 0.1$, respectively.

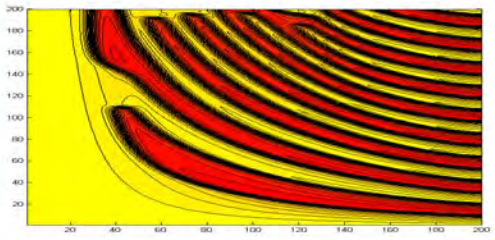


Figure 5. Space-time (i.e. j-t-) plot of the delayed chain (8) for $n = 200$, $\Delta = 0.002$, $d = 2.5$, $\tau = 0.08$

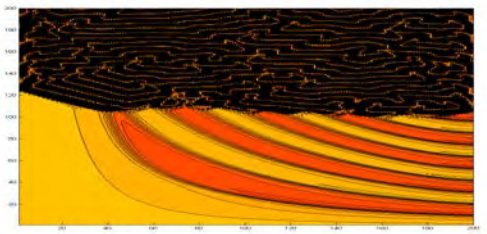


Figure 6. Space-time (i.e. j-t-) plot of the delayed chain (8) for $n = 200$, $\Delta = 0.002$, $d = 2.5$, $\tau = 0.09$

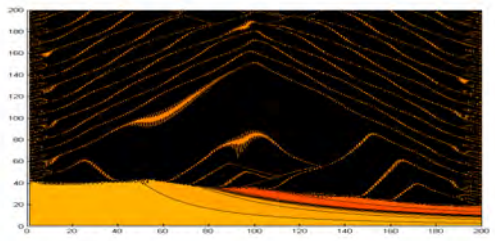


Figure 7. Space-time (i.e. j-t-) plot of the delayed chain (8) for $n = 200$, $\Delta = 0.002$, $d = 2.5$, $\tau = 0.1$

Whereas in Figure 5 we see a behavior which qualitatively resembles the behavior in the non-delayed case shown in Figure 2, we see in Figures 6 and 7 that after a certain time the delays lead to a distinctly different dynamics of the oscillator chain. Even more than in the non-delayed case, many of these phenomena still defy an analytical explanation.

3. APPLICATION IN MAX

3.1 Implementation

In order to experiment in real time we implemented a chain of $n = 50$ van der Pol oscillators in Max with a visual representation of the oscillator values depending on time and the number of the oscillators and a representation of the frequencies depending on time. Figure 8 shows the main features of the Max patch `smc16_vdp.maxpat` with the external `smc16_vdp_chain` and two plots. The inputs for the external are the frequency in Hz, ω/sr, the nonlinearity μ, the delay in samples, the coupling strength d, the amount of deviation of omega from one oscillator to the next (above called detuning, `d_omega`) and the random range of the omegas (`rand_omega`). The time step is the sample period. The plots show the oscillation values over time as gray level (left) and the frequencies over time (right).

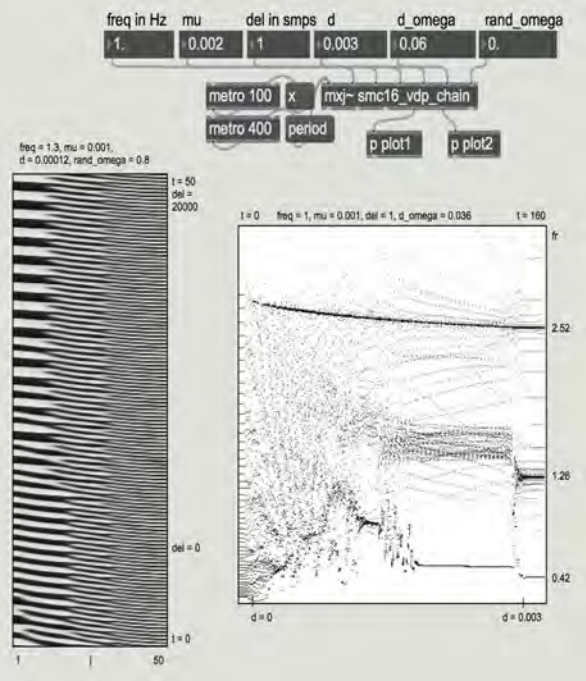


Figure 8. Simplified version of the Max patch `smc16_vdp.maxpat`

While for the production of the figures in section 2 we used pre-implemented methods, we will now show explicitly how to obtain discrete systems from the differential equations: first by the Euler method used in Neukom's studies 21.1-21.9 and then the classical Runge-Kutta method implemented in the Max-patch `smc16_vdp.maxpat` which we used to produce the following figures. The following Java code samples are taken from the perform routine of the above mentioned mxj~ external `smc16_vdp_chain`. The external and the Max patch can be downloaded from [18].

The implementation of Euler's Method for a single van der Pol oscillator is straightforward, the code is short and fast and with the sample period as time step quite precise [11]. First the acceleration `a` is calculated according to the differential equation above (1), using the nonlinearity `mu`. Then the velocity `v` is incremented by the acceleration times dt and displacement `x` by velocity times dt ($dt = 1$).

```
a = (- c*x + mu*(1 - x*x)*v);
        // with c = (frequency*2*Pi/sr)^2
v += a;
x += v;
```

The classical Runge-Kutta method (often referred to as RK4) is a fourth-order method. The values x and v of the next sample are approximated in four steps. The following code sample from the mxj~ external `smc_vdp` shows the calculation of the new values x and v using the function `f_` which calculates the acceleration.

```
double f_(double x, double v){return - c*x + mu*(1 -
    x*x)*v;}

k1 = f_(x, v);
l1 = v;
k2 = f_(x+l1/2, v+k1/2);
l2 = v+k1/2;
k3 = f_(x+l2/2, v+k2/2);
l3 = v+k2/2;
k4 = f_(x+l3, v+k3);
l4 = v+k3;
a = (k1 + 2*k2 + 2*k3 + k4)/6;
v += a;
x += (l1 + 2*l2 + 2*l3 + l4)/6;
```

The next code sample shows how a coupling with delay in a chain of n oscillators is implemented. Acceleration `a`, velocity `v`, displacement `x` and nonlinearity `mu` are stored in arrays of length n. In this implementation the coupling strength is the same for all connections. The oscillators are coupled by the difference of their own and the delayed velocity of their neighbors. The delay of the velocities is realized with n circular buffers by the two-dimensional array $delv[n][dmax]$ with length n times the maximal length $dmax$ of used delay. The positions where the velocities are written into and read out of the buffers are called pin and $pout$, respectively. The additional parts of the acceleration term due to the coupling conditions of the first and last oscillators of the chain are

```
d*(delv[1][pout] - v[0]);
d*(delv[n-2][pout] - v[n-1]);
```

and for the other oscillators

```
d*(delv[k+1][pout] + delv[k-1][pout] - 2*v[k]); // k = 1
    to n - 2
```

The inputs for the external `smc16_vdp_chain` are the

frequency in Hz, the nonlinearity μ, the delay in samples, the coupling d, the amount of deviation of omega from one oscillator to the next (the detuning d_omega, cf. (3)) and the random range of the omegas ($rand\_omega$, cf. (5)). From the input frequency we calculate `omega_0` of a corresponding linear oscillator (`mu` $= 0$). (`omega_0` $= (frequency * 2 * Pi/sr)^2$). From `omega_0` we get the individual omegas of the n oscillators either by exponentially or randomly distributing them depending on the input variables `d_omega` and `rand_omega`.

```
omega[k] = omega_0*(1 + d_omega*k); // k = 0 to n − 1

omega[k] =
    omega_0*(1+rand_omega*((float)(Math.random()−0.5f)));
    // k = 0 to n−1
```

We calculate the position of the index $pout$ for reading out the delayed velocities from the position of the index pin for writing the current velocities and the input delay del:

```
pout = pin − del;
```

For the following experiments the n factors μ (nonlinearity) and d (coupling) are identical.

In the visual representation of the oscillator values the x-axes represents the chain with the 50 oscillators and the y-axes the time in seconds. Every 100 ms a message triggers the output of the current oscillator values. The values are interpreted as gray level between -1 and 1.

In the visual representation of the evolution of the frequencies the x-axes represents the time and the y-axes the frequencies. Every 400 ms a message triggers the output of the current length of the periods of the oscillations.

3.2 Experiments

In a series of experiments we investigated chains with random and exponential distribution of the frequencies of the oscillators. We varied the delay and the coupling. Figure 9 shows the evolution of a chain with randomly distributed frequencies (`rand_omega` $= 0.8$). During the first 10 seconds there is no coupling, then the coupling grows linearly from 0 to 0.0002. With growing coupling more and more clusters appear where some oscillators synchronize their frequencies and by the time some clusters merge and the frequencies decrease. Figure 10 shows the same chain with the constant coupling 0.0001 and a growing delay. At the beginning of the simulation all oscillators were inactive. The first oscillator was excited by a small impulse. The excitation then propagates along the chain with growing delay. After a few seconds the clusters of synchronized frequencies appear. The phase differences between synchronized oscillators grow with the delay and the stripes in the figure become steeper. When the delay becomes longer than half of a period the delayed velocities of the neighbors of an oscillator have opposite sign to the velocity of this oscillator. As a result neighbors become out of phase and the difference of the velocities and hence the acceleration increase. In Figure 10 this happens at a delay of about 20000 samples where checked pattern begin to dominate; compare this uppermost pair of Figure 10 to the upper parts of Figures 6 and 7.

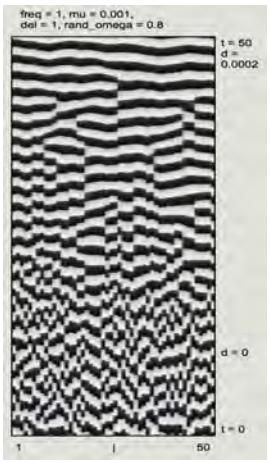


Figure 9. Randomly distributed frequencies

Figure 10. Constant coupling, growing delay

Figure 11 shows the evolution of a chain with exponentially distributed frequencies (`d_omega` $= 0.06$, cf. (4)). During the first 8 seconds there is no coupling, then the coupling grows linearly from 0 to 0.004. With growing coupling more and more oscillators synchronize and clusters merge. The low frequencies increase (the stripes on the left side of the figure become smaller) and the high frequencies decrease. The frequencies of the regions become stable and harmonic $0.42 : 1.26 : 2.52 = 1 : 3 : 6$ (see Figure 11). Figure 12 shows the same chain with the constant coupling 0.002 and a growing delay. After a few seconds the same clusters of synchronized frequencies appear as in the upper part of figure 11. The phase differences between synchronized oscillators grow with the delay and the stripes in the figure become steeper. All the frequencies decrease clearly.

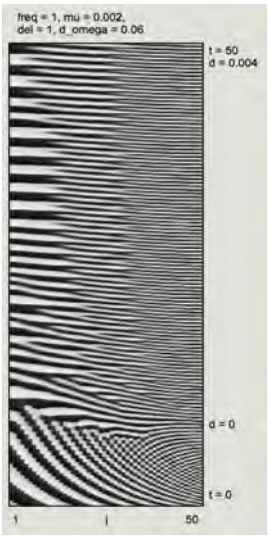


Figure 11. Exponentially distributed frequencies

Figure 12. Constant coupling, growing delay

Figure 13 shows the evolution of the frequencies with growing coupling strength.

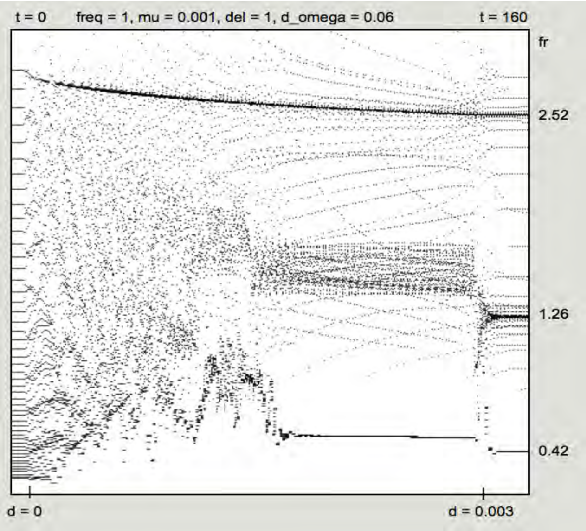


Figure 13. Frequencies in a van der Pol chain depending on growing coupling

4. MUSICAL APPLICATIONS

In Neukom's 8-channel studies 21.1-21.9 eight van der Pol oscillators are arranged in a circle and produce the sound for the eight speakers. Each of these oscillators has variable parameters frequency, nonlinearity and a gain and is coupled with his neighbors by variable coupling factors and delay times in both directions. Two additional chains of eight van der Pol oscillators produce control functions which are used for amplitude and frequency modulation. If the frequencies of the oscillators are lower than about 20 Hz the modulations produce pulsations and vibratos. Depending on the coupling strength and the delay some or all pulsations and vibratos synchronize their frequencies. The relative phase which is not audible in audio range plays an important role in the sub-audio range: the pulsations of the single sound sources can have the same frequency but be asynchronous in a rhythmic sense but with growing coupling strength they can produce regular rhythmic patterns, can be exactly in or out of phase.

5. FURTHER INVESTIGATIONS

There are many possibilities for future investigations in this area, e.g. to analytically explain the phenomena described numerically, introduce distinct values for the forward, backward, and self-delays, formulate and investigate a precise model for time- (or state-) dependent delays. Even in the non-delayed case, many open questions remain, such as the precise dependence of the synchronization tree on the various parameters of the model. Moreover, we only considered one-dimensional chains with nearest-neighbor-interactions; other interaction potentials or oscillator topologies could lead to other dynamical phenomena.

6. REFERENCES

[1] A. Pikovsky, M. Rosenblum, and J. Kurths, *Synchronization*. Cambridge University Press, 2001.

[2] L. Peltola, C. Erkut, P. R. Cook, and V. Välimäki, "Synthesis of hand clapping sounds," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 15, no. 3, pp. 1021–1029, 2007.

[3] C. Huygens, *Horologium Oscillatorium*. Apud F. Muguet, 1673.

[4] B. van der Pol, "Theory of the amplitude of free and fofrced triode vibration," *Radio Rev.*, vol. 1.

[5] E. V. Appleton, "The automatic synchronization of triode oscillators,," *Proc. Cambridge Phil. Soc.*, vol. 21, no. 231.

[6] A. Henrici and M. Neukom, "Synchronization in networks of delayed oscillators," in *Proceedings of the 42th Computer Music Conference ICMC, 12. - 16. September 2016, Utrecht, Netherlands, 2016*.

[7] S. H. Strogatz, *Sync: The Emerging Science of Spontaneous Order*. Hyperion Press, 2003.

[8] A. Arenas, A. Diaz-Guilera, J. Kurths, Y. Moreno, and C. Zhou, "Synchronization in complex networks," *Physics Reports*, vol. 469, no. 3, pp. 93–153, 2008.

[9] J. Pantaleone, "Synchronization of metronomes," *Am. J. Phys*, vol. 70, no. 10, pp. 992–1000, 2002.

[10] https://www.youtube.com/watch?v=5v5eBf2KwF8, accessed: 2016-04-13.

[11] M. Neukom, "Applications of synchronization in sound synthesis," in *Proceedings of the 8th Sound and Music Computing Conference SMC, 6. - 9. July 2011, Padova, Italy, 2011*.

[12] ——, *Signals, Systems and Sound Synthesis*. Peter Lang, 2013.

[13] G. V. Osipov, J. Kurths, and C. Zhou, *Synchronization in Oscillatory Networks*. Springer-Verlag, 2007.

[14] F. M. Atay, "Van der pol's oscillator under delayed feedback," *J. Sound and Vibration*, vol. 218, no. 2, pp. 333–339, 1998.

[15] K. Hu and K. Chung, "On the stability analysis of a pair of van der pol oscillators with delayed self-connection, position and velocity couplings," *AIP Advances*, vol. 3, p. 112118, 2013.

[16] M. Lakshmanan and D. Senthilkumar, *Dynamics of Nonlinear Time-Delay Systems*. Springer-Verlag, 2010.

[17] F. M. Atay, *Complex Time-Delay Systems: Theory and Applications*. Springer-Verlag, 2010.

[18] https://www.zhdk.ch/index.php?id=icst_downloads, accessed: 2016-04-10.

Movement Sonification of Musical Gestures: Investigating Perceptual Processes Underlying Musical Performance Movements

Jesper Hohagen
jesper.hohagen@uni-hamburg.de

Clemens Wöllner
clemens.woellner@uni-hamburg.de

Institute of Systematic Musicology, University of Hamburg

ABSTRACT

Truslit (1938) developed a theory on the gestural quality of musical interpretations. Self-other judgment paradigms of visual point-light movements allow elucidating action-perception coupling processes underlying musical performance movements as described by Truslit. Employing movement sonification with a continuous parameter mapping approach may further show parallels between the audio information of music, physical movements, and audio information based on sonified movement parameters. The present study investigates Truslit's hypothesis of prototypical musical gestures by comparing free movements and movements following detailed instructions recorded by a 12-camera optical motion capture system. The effects of watching these movements and listening to the sonification were tested within a multimodal self-other recognition task. A total of 26 right-handed participants were tracked with a motion capture system while executing arm movements along with Truslit's (1938) original musical examples. The second experimental part consisted of a multimodal self-other perception judgment paradigm, presenting sequences to the same participants (matched with those of two other participants, unbeknown to them) under four different conditions. Signal detection analyses of the self-other recognition task addressed judgment sensitivity by calculating for individual participants. While self-recognition was successful for visual, audiovisual and still image examples, movement sonification did not provide sufficient detail on performer's agency. Nevertheless, a number of relevant sonification parameters is discussed.

1. BACKGROUND

Alexander Truslit proposed a theory of gestalt and movement that highlights the gestural quality of musical interpretations [1]. In his work he attempted to investigate the connection between the inner shape and motion of music and the perceptual processes of listeners while executing movements along with the music. Thereby, Truslit posed questions of the prototypicality of musical gestures. Since Bruno Repp's [2] synopsis of Truslit's work, researchers have increasingly developed methods and paradigms to study listeners' responses to musical gestures. Truslit himself presented movement graphs to selected musical pieces, based on dynamic and agogic information, assuming that these motion trajectories are valid intersubjectively. In contrast to Becking [3], who assumed that there are distinct and stable motor pulses in the works of different composers, Truslit hypothesized that listeners can learn to *feel* and then reproduce the shape and motion lying inside the music. However, there are still open questions concerning the common factors of musical and movement parameters, for example on how individuals perceive musical motion while they perform movements related to the music. Further empirical examinations are required to test this theory of a prototypicality of musical movements on a descriptive and comparative level.

In order to reach insights into the perceptual processes of music listeners and performers, asking for self-other judgments of visual point-light movements appears to be a promising method [4, 5]. Beside such a perception paradigm allowing the study of action-perception coupling [6], movement sonification may provide listeners with intuitive feedback on musical movement features. So far, movement sonification has mainly been applied in artistic performances [7] or sport and rehabilitation science [8, 9]. To our knowledge, our study is the first to systematically investigate the motion of Truslit's gestures with sonification of gestures and a self-other perception paradigm.

2. AIMS AND HYPOTHESES

In a first study (Table 1), we investigated Truslit's hypothesis of prototypical musical gestures by comparing free movements to Truslit's original sound examples with movements following a visual presentation and detailed verbal instructions. The second study tested the effects of watching point-light displays and listening to the sonification of movements with a multimodal self-other judgment paradigm.

Along with various analogous experimental tasks, we expect differences in expression of the movements before and after instruction. Moreover, we assume that this variation is higher in musically experienced participants in comparison with non-musicians, according to the for-

mers' capacity to receive and process musical material and transfer it to appropriate movements.

Referring to results of established self-recognition tasks, we assumed that self-identification of visual displayed movements would be above chance. We expected higher scores of self-recognition for the unconstrained movements compared to post-instruction movements and a better performance in judging self and others movements by musicians in comparison with non-musicians.

Study	Conditions	Stimuli
Study 1 *(Performance task)*	Block 1: free condition	1. Wagner – *Gebet der Elisabeth* 2. Verdi – *Celeste Aida*
	Block 2: instruction condition	*Broken Chord C major, staccato* – bassoon
	Block 3: after instruction condition	Same as in 1st block
Study 2 *(Perception task)*	1. Visual (v)	Animation of visual point-light displays
	2. Auditory (a)	Movement data sonification
	3. Still image (si)	Still Image of movement trajectory graph
	4. Audio-visual (av)	Animation of visual point-light displays & sonification

Table 1. Overall research design

3. STUDY 1

3.1 Participants

In Study 1, a total of 26 right-handed participants (age: M=27.35, SD=4.06; 30,8% female, 13 musicians) were recruited to take part in the performing sessions.

3.2 Materials

23s cuttings of three selected original Truslit recordings (1. *Gebet der Elisabeth* – Richard Wagner, 2. *Celeste Aida* – Giuseppe Verdi, 3. *Mondnacht* – Robert Schumann) were presented during the first and the third blocks of the recording sessions. In addition, another 7s-short original Truslit piece (*Broken Chord C major, staccato* – bassoon) was played within the instruction part during the second block. While listening to the *broken chord* piece, an original Truslit drawing of a set definition movement trajectory was presented on a screen (see Fig. 1).

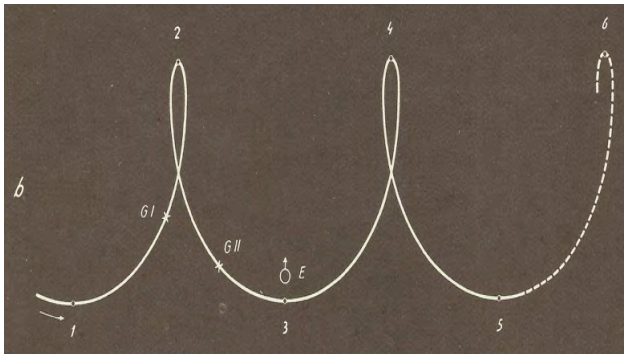


Figure 1. Original Truslit movement graph of *Broken Chord C major, staccato* – bassoon, used for visual instruction

3.3 Design and procedure

Performing sessions in Study 1 were divided in two main blocks, each with three different movement trials linked to three different musical stimuli. In between these two recording blocks, we integrated a shorter instruction block. We had a 2x2 repeated-measured design with two within-participant factors of instruction (Block 1: free movements along with three different musical stimuli; Block 3: after instruction, movements following the same musical stimuli as in the first block) and musical excerpt (Wagner; Verdi).

Participants' movements were tracked by a 12-camera 3D OptiTrack® motion capture system. For the recording sessions of the conducting-like gestures such as in Truslit (1938), a single marker was placed around the index finger of participants' right hand.

In each trial of the first block (free condition) participants received the instruction to "follow with your right arm freely the melody of the song you listen to". After a practice trial, participants performed three trials along with three original Truslit songs. Each time, participants first listened to the song before moving their arm to the melody of the song in order to become familiar with the style of the musical pieces. The three songs were presented randomly across participants to control order effects.

As part of the Block 2, participants first listened to 7s pieces of a broken chord sequence (2x) before they following the melody (2x) as explained. Second, participants listened to the same piece three times consecutively while looking at the appropriate Truslit trajectory graph (see Fig. 1) on a screen. The third part of this second session was a replication of the first one, but with the verbal instruction beforehand to "follow the melody with the index finger right in the way you saw it on the screen".

The third block (after instruction condition) was a replication of the first block without any additional instructions. Again, the musical pieces were played in randomized order.

3.4 Results

In order to assess differences between Block 1 (free) and Block 3 (after instruction) as well as between the two

musical excerpts, we analyzed the averaged global measures of their index finger movement lines in terms of movement velocity, acceleration, jerk and cumulative distance (for descriptive analysis of movement parameters see Table 2).

Movement parameters	Conditions	Averaged global measures
Velocity	Free - Wagner	M=0.023m/s
	After inst. - Wagner	M=0.017m/s
	Free - Verdi	M=0.021m/s
	After instr. - Verdi	M=0.020m/s
Acceleration	Free - Wagner	M=-0.005m/s²
	After inst. - Wagner	M=-0.005m/s²
	Free - Verdi	M=-0.006m/s²
	After instr. - Verdi	M=-0.010m/s²
Jerk	Free - Wagner	M=0.033m/s^3
	After inst. - Wagner	M=0.003m/s^3
	Free - Verdi	M=-0.048m/s^3
	After instr. - Verdi	M=-0.076m/s^3
Cumulative distance	Free - Wagner	M=1.995m
	After inst. - Wagner	M=1.959m
	Free - Verdi	M=2.815m
	After instr. - Verdi	M=2.820m

Table 2. Averaged global measures of movement parameters

The Schumann song was not taken into consideration for statistical data analysis due to large gaps in some motion capture data.

A 2x2 repeated-measures ANOVA indicated significant differences between the velocity values of the Blocks 1 and 3 regarding the factor Instruction ($F\{1, 25\}=5.40$, $p=.029$, $\eta^2=.177$), indicating that participants moved more quickly in the free condition compared to the post-instruction block.

When comparing jerk values, we found a significant difference between the Wagner and Verdi piece, ($F\{1, 25\}=8.56$, $p=.007$, $\eta^2=.255$). The fact that participants jittered more during the Wagner song can be explained my musical parameters, i.e. the melody line of the Wagner piece is less active, so participants may have struggled to hold the line with their finger and started to shake.

Furthermore, there are highly significant differences in cumulative distance values between Verdi and Wagner movements ($F\{1, 25\}=20.17$, $p<.001$, $\eta^2=.447$). As observed for the results above, this difference shows again the effect of musical features based on the more active melody line of the Verdi piece compared to Wagner, leading to a higher distance travelled for Verdi movements.

No significant interaction effects between the factors musical excerpt and instruction were found. Furthermore, participants were relatively consistent in their movement styles irrespective of the Truslit-based instructions. In a similarly vein, analysis of covariance indicated no effects of musical experience or preferences ratings for the music on movement characteristics**.**

4. STUDY 2

4.1 Participants

In Study 2, 23 (age: M=27.43; SD=4.29; 30.4% female, 11 musicians) out of the 26 participants from the first study took part in the perception task experiment.

4.2 Materials

The multimodal self-other recognition task consisted of four different display conditions, based on the movement trajectory data recorded within the performing trials of Study 1.

4.2.1 Animated visual point-light displays (v)

For preparing the first condition of the self-other perception paradigm, we used the data of movement trajectories of the index finger marker from the performing sessions of the first study. For each participant, we created four 2D 10s normalized video clips with point-light displays of the finger marker (end points of data on both axes were standardized, in order to avoid recognition skills based on maximum movement amplitude) from the original 23s excerpts of Study 1 (1st clip: free – Wagner; 2nd clip: free – Verdi; 3nd clip: post-instr. – Wagner; 4th clip: post-instr. – Verdi). Video animations were created by using the MoCap-Toolbox [10] for Matlab®.

4.2.2 Sonifications of movement data (a)

The second condition for the perception task contained again four 10s clips per participant, but this time data of movement trajectories was matched with a continuous auditory feedback. For preparing these sonification sequences the continuous parameter mapping method [11] was applied, i.e. vertical position data (y-axis) of finger movements were matched with the pitch of a continuous synthesizer and horizontal position data (x-axis) were matched with stereo panning of the same sound. With this sonification mapping participants just "heared" their own movements captured in the first study, and during the presentation of the sonification clips the screen was black. Sonification sequences were programed by using Python® and the audio synthesis programming language SuperCollider.

4.2.3 Still Images of movement trajectories (si)

In the third perception condition, the same animation process as in the first condition was applied, but we created four 10s still images of the movement trajectories for each participant, comparable with the movement graphs of Truslit. For an exemplary trajectory graph of one par-

ticipant see Fig. 2 (free – Wagner) and Fig. 3 (after inst. – Wagner).

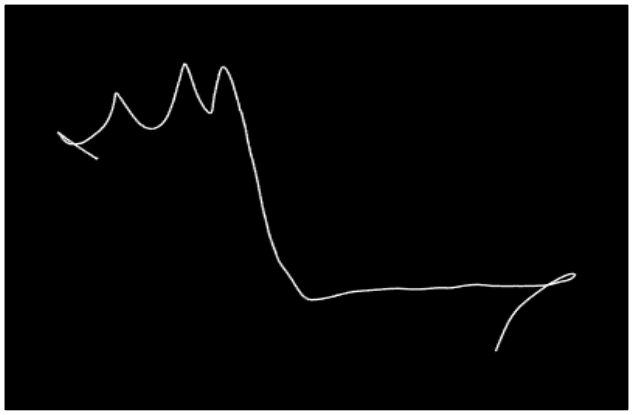


Figure 2. Exemplary trajectory graph for free Wagner movements (10s) of index finger (Block 1)

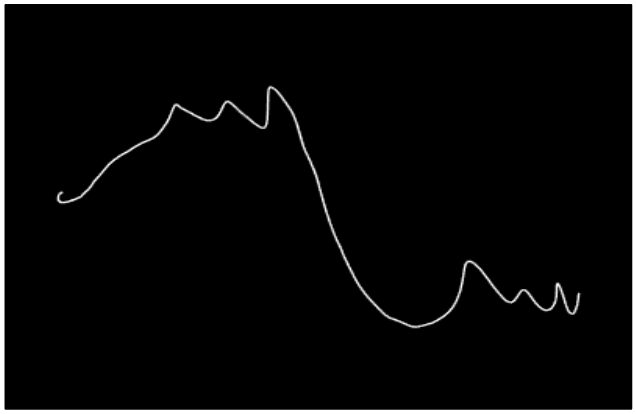


Figure 3. Exemplary trajectory graph for post-instruction Wagner movements (10s) of index finger (Block 3)

4.2.4 Visual point-light and auditory displays (av)

The fourth condition was a combination of the first (v) and second (a) perception condition, i.e. participants watched the visual point-light displays on a screen while listening to the sonification of the same movement data at the same time – again 4x 10s clips.

4.3 Design and procedure

9 months after the recording sessions of Study 1, 23 out of the 26 participants took part on a perception task experiment. We choose this long interlude time to avoid movement memory effects, which means that participants would have recognized their own movements based on their memory skills.

Within the self-other recognition paradigm, the multimodal movement displays of one participant were matched with sequences of two other participants by height and sex [12]. This method is useful to avoid recognition effects based on body information that can be interpreted from the video clips. Overall, each participant watched and listened to four 10s sequences of oneself and four sequences of the two other participants unbeknown to them. Every clip was presented twice across the four conditions (12 clips x 4 conditions x 2 presentations of each clip = 96 clips, divided in four condition blocks), thus, the total test time was around 60 minutes.

After watching or listening the 10s sequences, participants judged whether they had perceived their own movements or those of someone else. In addition, they answered how sure they were in their judgments and how expressive as well as how fluent the movements appeared to them (on 7-point Likert Scales from 1 - "very secure/expressive/fluent" to 7 - "not at all").

Participants were not informed whether the sequences displayed free or post-instructed respectively Wagner or Verdi movements. The two original Truslit excerpts were not played while the participants watched the point-light displays of their movement performances.

4.4 Results

Analyses of the self-other recognition task addressed judgment sensitivity by calculating d-prime (d') scores for individual participants, i.e. we substract z-transformated false alarm rates (participants incorrectly assume that they perceive the displayed movement as their own) from hit rates (correct self-recognition). We assumed that self-identification of visual and auditory displayed movements is above chance in all conditions. One-sample t-tests revealed that self-recognition was successful in three conditions: v ($t\{22\}=2.21$, $p<.05$); si ($t\{22\}=2.45$, $p<.05$) and av ($t\{22\}=2.46$, $p<.05$). These results show the ability to recognize one's own movements even in a perception task in which body information was strongly reduced, indicating that kinematic information sufficed for participants' recognition accuracy.

No significant results were found for the auditory display condition, indicating that participants could not map the movement sonification intuitively with the shape of their movements.

Within the self-other recognition task, musicians scored significantly higher in the visual ($t\{21\}=2.29$, $p<.05$) and audiovisual conditions ($t\{21\}=2.31$, $p<.05$) compared with non-musicians. Thus, musicians in this study possessed advanced skills in recognizing their own music-related movements potentially based on enhanced action-perception-coupling for these musical tasks.

5. CONCLUSIONS

While there were large inter-individual differences in the movement trajectories of participants in Study 1, analyses revealed a high consistency in the repeated-measures condition, so that individuals performed comparable movements across trials. These results were unexpected, considering the clear movement instructions in Block 2 based on Truslit's motion shapes. However, results of significant differences between performance conditions (free – after instruction) indicates small effects of moving intuitively on movement velocity. Furthermore, musical characteristics seem to influence movement execution, so

that participants travelled longer resp. moving more while listening to the song with a complex melody line, that is Verdi's *Celeste Aida*. On the other hand, we see typical movement characteristics in terms of jitter, while moving to a melody with longer tone sequences. Thus, Wagner's *Gebet der Elisabeth* leads to a significantly higher jerk compared to Verdi. In further studies, we will focus on such correlations between musical features and movement parameters, so we will include more spatial and temporal parameters of the recorded movement trajectories as well as musical and acoustical analysis of the original Truslit samples.

Results of the self-other recognition task indicate a common perceptual basis that is grounded in human movements and lies beyond individual percepts of music, but just in terms of visual perception processes. Musicians tend to recognize their movements more often correctly compared to non-musicians, possibly showing advanced musical perception processes due to their expertise in moving while making music. The sonification used in this study did not lead to a self-recognition above chance. Further methods will be employed an auditory display method that tries to get on a deeper layer of perceptional processes while listening to sonification of movement data. Therefore, an evaluation study of different sonification mappings, sounds and styles appears to be a promising approach.

6. REFERENCES

[1] A. Truslit, Gestaltung und Bewegung in der Musik. Ein tönendes Buch vom musikalischen Vortrag und seinem bewegungserlebten Gestalten und Hören. Vierweg, 1938.

[2] B. Repp, "Music as Motion: A Synopsis of Alexander Truslit´s (1938) Gestaltung und Bewegung in der Musik. Psychology of Music, 1993, pp. 48-72.

[3] G. Becking, Der musikalische Rhythmus als Erkenntnisquelle, Benno Filser, 1928.

[4] V. Sevdalis, and P.E. Keller, „Cues for self-recognition in point-light displays of actions performed in synchrony with music", Consciousness and Cognition, 2010, pp. 617-626.

[5] C. Wöllner, „Self-recognition of highly skilled actions: A study of orchestral conductors", Consciousness and Cognition, 2012, pp. 1311-1321.

[6] W. Prinz, „Perception and action planning", European Journal of Cognitive Psychology, 1997, pp. 1-20.

[7] A. Renault, C. Charballier, and S. Chagué, „3dinmotion – a Mocap Based Interface for Real Time Visualisation and Sonification of Multi-User Interactions", in NIME International Conferences on New Interfaces for Musical Expression, 2014, pp. 495-496.

[8] N. Schaffert, Sonifikation des Bootsbeschleunigungs-Zeit-Verlaufs als akustisches Feedback im Rennrudern, Logos, 2011.

[9] A.O. Effenberg, U. Fehse, and A. Weber, „Movement Sonification: Audiovisual benefits on motor learning", in B.G. Bardy, J. Lagarde, D. Mottet (eds.), BIO Web of Conferences. The International Conference SKILLS, 2011, pp. 1-5.

[10] B. Burger, and P. Toiviainen, „MoCap Toolbox – A Matlab toolbox for computational analysis of movement data", in R. Bresin (Ed.), Proceedings of the 10th Sound and Music Computing Conference, (SMC), 2013, pp. 172-178.

[11] F. Grond, and J. Berger, „Parameter mapping sonification", in T. Hermann, A. Hunt, J.G. Neuhoff (eds.), The Sonification Handbook, 2011, pp. 363-398

[12] F. Loula, S. Prasad, K. Harber, and M. Shiffrar, „Recognizing people from their movement, in J. of Exp. Psych., 2005, pp. 210-220.

PRIMARY-AMBIENT EXTRACTION IN AUDIO SIGNALS USING ADAPTIVE WEIGHTING AND PRINCIPAL COMPONENT ANALYSIS

Karim M. Ibrahim
Nile University
k.magdy@nu.edu.eg

Mahmoud Allam
Nile University
mallam@nu.edu.eg

ABSTRACT

Most audio recordings are in the form of a 2-channel stereo recording while new playback sound systems make use of more loudspeakers that are designed to give a more spatial and surrounding atmosphere that is beyond the content of the stereo recording. Hence, it is essential to extract more spatial information from stereo recording in order to reach an enhanced upmixing techniques. One way is by extracting the primary/ambient sources. The problem of primary-ambient extraction (PAE) is a challenging problem where we want to decompose a signal into a primary (direct) and ambient (surrounding) source based on their spatial features. Several approaches have been used to solve the problem based mainly on the correlation between the two channels in the stereo recording. In this paper, we propose a new approach to decompose the signal into primary and ambient sources using Principal Component Analysis (PCA) with an adaptive weighting based on the level of correlation between the two channels to overcome the problem of low ambient energy in PCA-based approaches.
Key words: Audio Source Separation, Primary/ambient Separation, Surrounding Sound Systems, Upmixing.

1. INTRODUCTION

Currently, most audio recordings are available as 2-channel stereo recordings. For a long time, this has been considered sufficient to give the listener a pleasant experience. However, with new sound systems that give a better sense of surrounding and enclosing atmosphere, older recordings fail to utilize the capabilities of these new systems. Thus, it is important to develop methods of extracting additional spatial information from these recordings to enhance the experience of listening to them: this process is called upmixing [1, 2]. One approach is applying audio source separation to extract the original sources from the mixture, which are then rendered for the new playback system [3]. An important distinction between the different audio sources that can be used as a base for separating the sources is the ability to localize the sound sources. Separating sources based on their directional and diffuse features can be used in upmixing to create an immersive feeling.

5.1 surround systems [4] are an example of a multi-channel sound system commonly used in home theaters that are often used to play stereo recordings. A practical method of upmixing the stereo sound to the 5.1 system is by separating the primary (localizable) and ambient (non-localizable) sources and playing the primary sources on the two front channels to recreate the direct sources as it was intended in the original recording while playing the ambient sources on all channels to give a better feeling of surround sound.

Such applications call for advanced audio source separation methods. Hence, such methods have increasingly gained attention in the research community. Audio source separation can generally be categorized into two main challenges: blind audio source separation (BASS), where the goal is to extract the different sound sources in the mix, and primary-ambient extraction (PAE), where the goal is to separate between primary (direct) sources and ambient (diffuse) sources.

Several approaches have been proposed to extract the primary and ambient sources from a mixed-down recording. A commonly used approach is using Principal Component Analysis (PCA) as in [5, 6], which is investigated in detail later in this paper as it is the basis for the proposed approach. A different approach for the problem is using the least square method to estimate the primary and ambient sources as proposed by Faller in [7] by minimizing the errors between the extracted signals and the original stereo input.

In Avendano's work [8], the approach is to calculate a band-wise inter-channel short-time coherence from the cross- and autocorrelation between the stereo channels which is then used as the basis for the estimation of a panning and ambiance index. In Kraft's approach [9], the proposed method is based on the mid-side decomposition of stereo signals where the two-channel recording is split into "mid" signal that captures the centered content of the recording and a "side" signal that captures the content panned to the left and right side.

The focus of this paper lies in developing a new technique for primary-ambient extraction in stereo signals and to introduce an evaluation method for PAE to compare between the different commonly used approaches and our new proposed method.

The paper is structured as follows: Section 2 explains the problem definition of audio source separation and primary ambient extraction, the possible application for these tech-

niques and the constraints for an ideal extraction.

Section 3 explains our proposed method to improve the separation based on Principal Component Analysis (PCA). Finally, Section 4 shows the evaluation between the proposed method and the previous methods from the literature.

1.1 Notation

The convention in this paper is to express signals in the time domain in lower case letter as x, while signals in the STFT domain are in upper case as X. Scalar variables are expressed in normal italic font as X while column vectors are expressed in bold italic font as $\boldsymbol{X}$ and matrices are expressed in bold non-italic font as $\mathbf{X}$.

Table 1 shows the commonly used symbols in this paper:

$\boldsymbol{x}$	Mixed stereo signal
$\boldsymbol{x}_l$, $\boldsymbol{x}_r$	left and right channels of a sound mixture
$\boldsymbol{p}_l$,$\boldsymbol{p}_r$	Left and right primary components
$\boldsymbol{a}_l$,$\boldsymbol{a}_r$	Left and Right ambient components
n	Discrete time index
m	Frequency index
k	Frame index
w_{pl},w_{pr}	weighting factor of the primary source
$\boldsymbol{v}$	Normalized unit vector of 1^{st} Principal component

Table 1: Symbols used in this paper

2. PRIMARY-AMBIENT EXTRACTION

One of the key characteristics in spatial audio is whether an audio source is localizable or not. A localizable source is perceived as coming from a certain direction and the listener can determine this direction, also called primary or directional source. A non-localizable source is perceived as a surrounding sound, coming from all around, also called an ambient or diffuse sound. Ambient sources usually describe the surrounding atmosphere of the recording . Methods for separating these two types of sources have been receiving increasing attention for applications such as upmixing [10, 11], multichannel format conversion and headphone reproduction [12, 13].

2.1 Signal model for PAE

When approaching the problem of primary-ambient extraction, we consider the input signal as a mix of two sources; a primary and an ambient source. In this paper, we only approach the problem of separating the mixture of a stereo signal.

Stereo recordings consist of two channels that contain both the primary and ambient sources mixed together and the goal is to separate them. The signals can be expressed as follows:

$$x_l[n] = p_l[n] + a_l[n] \quad (1)$$

$$x_r[n] = p_r[n] + a_r[n] \quad (2)$$

where x_l, x_r are the left and right channels of the stereo recording respectively, p_l, p_r are the primary component in each channel, a_l, a_r are the ambient component and n is the time index of the discrete signals.

Most PAE approaches are applied in the STFT domain as it is safer to assume there is only one primary source and one ambient source in each frequency-frame sub-band. The signals are expressed then in the form:

$$X_l[m,k] = P_l[m,k] + A_l[m,k] \quad (3)$$

$$X_r[m,k] = P_r[m,k] + A_r[m,k] \quad (4)$$

where m, k are the frame and frequency index respectively.

2.2 Sound localization and human auditory system

To be able to precisely separate the primary and ambient sources, it is necessary to understand how the human auditory system works and how it determines the location of a sound source and then use the same characteristics in the separation process.

The human auditory system uses several cues to localize a sound source, including inter-channel time difference (ICTD), also referred to as inter-aural time difference (ITD), inter-channel level difference (ICLD), also referred to as inte-aural level difference (ILD), spectral information and correlation analysis [14].

A comparison between the two channels should be sufficient to extract the directional information of an audio source. The correlation between the two channels plays a significant role in determining the location of the source, i.e., an ambient source shows no correlation between the two channels, making it impossible for the human auditory system to determine the direction of the sound. Hence, calculating the correlation between the two channels is usually a necessary step in extracting the primary and ambient sources.

2.3 PAE applications: upmixing to 5.1 systems

A common application for PAE is upmixing from n to m channels, where $m > n$. Here, we explain how to use PAE in upmixing to one of the commonly used systems, the 5.1 surround system. By separating the primary and ambient sources using one of the PAE methods, the extracted sources are re-panned in a way that the left primary sources $p_l[n]$ are played on the front left and center channels, $x_{lf}[n]$ and $x_c[n]$ while the right primary sources are played on the front right and center channels $x_{rf}[n]$ and $x_c[n]$ and the ambient sources are played throughout the five speakers. This way, the directionality of the primary sources are kept as originally intended while the surrounding sound is enhanced by the ambient sources. Figure 1 shows the block diagram of the upmixing technique.

2.4 PAE assumptions

To accurately separate between the primary and ambient components, we need to define the constraints that achieve the right separation. By definition, the primary sources are localizable while the ambient sources are non-localizable.

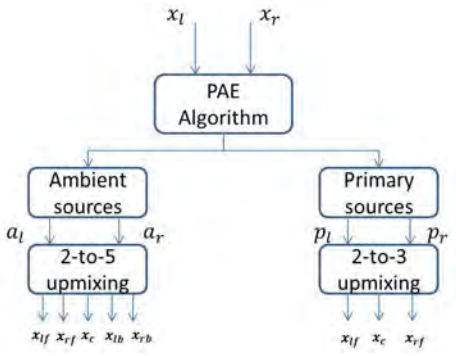


Figure 1: Block diagram of the stereo to 5.1 upmixing using PAE

To find a mathematical representation for this definition, we need to review the sound localizing process in the human auditory system mentioned in section 2.2. The key characteristic in localizing the sound sources is the correlation between the two signals reaching the left and right ears. In the case of a complete non-localizable diffuse source, the two signals are expected to be orthogonal in a way that the brain fails to detect any similarity between the left and right signals to extract location information. Similarly, primary sources are expected to be partially or fully correlated. Based on the representation of the stereo signal in equation (3) and assuming that the left and right primary components are P_l, P_r respectively, where P_l, P_r are vectors of adjacent STFT frames, Similarly the left and right ambient components are A_l, A_r, ω is the scaling factor between the primary components in the two channels due to ICLD and $\boldsymbol{A}^H$ is the Hermitian transpose of the vector $\boldsymbol{A}$, these constraints are defined according to [5] as:

1. The primary components are correlated

$$\boldsymbol{P}_l = \omega \boldsymbol{P}_r \tag{5}$$

2. The ambient components are orthogonal (fully uncorrelated)

$$\boldsymbol{A}_l^H \boldsymbol{A}_r = 0 \tag{6}$$

3. The ambient and primary components are orthogonal to each other

$$\boldsymbol{P}_l^H \boldsymbol{A}_l = 0 \qquad \boldsymbol{P}_r^H \boldsymbol{A}_r = 0 \tag{7}$$

4. The two ambient components have almost the same energy level

$$\boldsymbol{A}_l^H \boldsymbol{A}_l \approx \boldsymbol{A}_r^H \boldsymbol{A}_r \tag{8}$$

Figure 2 shows the assumed constraints between the different components.

2.5 PCA-Based PAE

Many of the approaches of PAE are based on the Principal Component Analysis (PCA) as in [5,6,15–18]. PCA is widely used since the common signal model assumes that the stereo signal is composed of primary sources that are

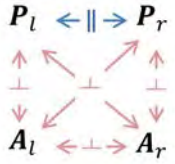


Figure 2: Constraints on the primary and ambient components

highly correlated and ambient diffuse sources. It is suitable to use a decomposition method such as PCA to extract the correlated primary sources and to assume the ambient sources are the residuals. The work in [15] is also based on the PCA but with an important modification, it takes into consideration the Inter-Channel Time Difference (ICTD) by using a time-shifting technique to improve the extraction of the primary sources.

One major drawback of methods based on PCA is the assumption that there is always a primary source in each frequency-frame sub-band and that it is never too weak. This is evident from the extraction of the primary sources as the first principal component. In case of absence of any primary sources, the method would still assign the first principal component, the one with the highest energy, to the primary source, which clearly produces a significant error in this particular case.

3. IMPROVING PCA-BASED APPROACH

As described in Section 2.5, the PCA-based approach has a number of drawbacks that impairs its accuracy. The solution we propose is to add an adaptive weighting to increase the amount of energy the ambient signal. The concept of adaptive weighting in PCA was previously introduced by Goodwin [19] with a different weighting scheme. The weighting we propose is based on the relation between the two channels of the signal in a way that supports the ambient extraction by detecting the level of presence of the primary sources. One way to do this is by considering the second dominant eigenvalue and comparing its value to the dominant eigenvalue. In the case of high correlation, the first (dominant) eigenvalue will be considerably larger than the second eigenvalue. In this case it would be safe to decompose the signal into primary and ambient components. However, in the case of having a more dominant ambient source, the ratio between the first and second eigenvalues will be relatively small.

The PAE using our weighting scheme is applied as follows:

1. We start with the original 2-channel signals, $x_l[n]$ and $x_r[n]$. We apply the STFT on the signals to get $X_l[m,k]$ and $X_r[m,k]$, where m is the frame index and k is the frequency index. We calculated the STFT using $\frac{3}{4}$ overlapping Hamming windows of Length 4096 samples, corresponding to a duration of 92.8 milliseconds at a sampling frequency of 44.1 kHz.

2. For each frequency-frame bin we define a vector with

the STFT values of the M adjacent frames:

$$\boldsymbol{X}_{l,r}[m,k] = \begin{bmatrix} X_{l,r}[m-M,k] \\ \vdots \\ X_{l,r}[m+M,k] \end{bmatrix} \quad (9)$$

For brevity, the index $[m,k]$ is dropped in the following equations.

3. The decomposition is then applied per frame-frequency index to extract the primary and ambient components in each frame-frequency using these two vectors as shown in Figure 3.

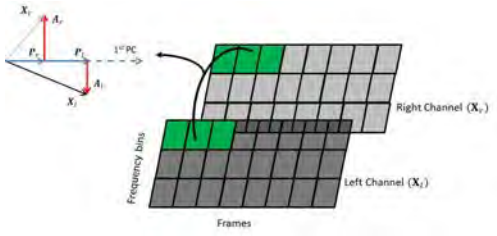


Figure 3: PAE using PCA

4. Using the Eigendecomposition on the covariance matrix $\mathbf{C}$ of the two vectors $\boldsymbol{X}_l, \boldsymbol{X}_r$, we get the normalized dominant Eigenvector $\boldsymbol{V}$ and the first two dominant Eigenvalues λ_1, λ_2.

5. Next we calculate the weighting factor ω based on the ratio between the two eigenvalues. The primary weights are defined as:

$$\omega = 1 - \frac{\lambda_2}{\lambda_1} \quad (10)$$

This weighting ensures that in cases of low correlation between the two channels, a higher weight is given to the ambient component.

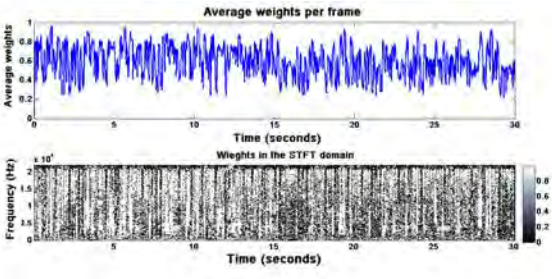


Figure 4: Sample of the weights extracted from an audio file

6. In order to enhance the ambient extraction, we define a threshold θ. The goal is to detect the cases where there is no strong presence of a primary source, so all the content is put into the ambient component. The primary component is still weighted by ω in case of passing the threshold to support extracting the ambient component.

$$\boldsymbol{P_{l,r}} = \begin{cases} \omega(\boldsymbol{V}^{\boldsymbol{H}}\boldsymbol{X_{l,r}})\boldsymbol{V}, & \omega > \theta \\ 0, & \omega < \theta \end{cases} \quad (11)$$

$$\boldsymbol{A_{l,r}} = \boldsymbol{X_{l,r}} - \boldsymbol{P_{l,r}} \quad (12)$$

where P_l, P_r, A_l, A_r are the primary and ambient components of the right and left channel respectively. Figure 4 shows an example of the weights extracted from an audio file.

7. Finally, two schemes can be used to extract the information from each frame; either to merge the extracted vectors by averaging them or to take out the center point of each vector as shown in figure 5. However, the results of both schemes are very similar, so we can use the "take center" scheme to reduce the computation.

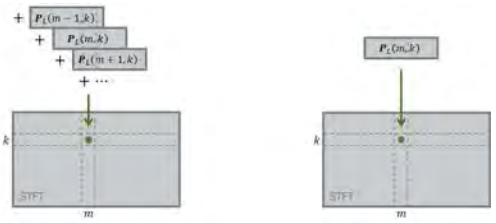


Figure 5: Different schemes of merging the output vectors

4. EVALUATION

Our objective evaluation is based on the work done in [20] which is intended to evaluate blind audio source separation (BASS). It can be used for primary-ambient separation, as well, by assuming that the mixture of sources is made out of only two sources, an ambient and a primary one. Ideally the extraction methods should output two sources that are identical to the originals ones. However, due to the limitations of the extraction methods, there is interference between the two sources.

In the following we would like to compare the following approaches:

1. PCA-based PAE without weighting, referred to as "PCA without weighting".

2. The weighted PCA method by Goodwin in [5, 6]. Referred to as (PCA Goodwin).

3. The extraction method by Avendano and Jot in [8]. referred to as (Avendano)

4. The modified PCA method with the weighting scheme proposed in this paper with two different threshold values: $\theta = 0.5$ and $\theta = 0.9$.

The evaluation was performed using two databases, one is made out of all ambient sources, consisting of strong ambient sources as sounds of crowd, forest, rain and echoes, and the second is made of all primary sources, consisting of strong primary sources as vocal recordings, solo instruments and dialogs. Each of the two data sets consist of 40 different recordings that are mixed together to compose 40 mixed recordings. We used the Matlab toolbox "BSS Eval" [21] for calculating the errors. The evaluation is as follows:

1. Mixing one ambient source with one primary source after normalizing the two of them, by ensuring the highest energy level of the two sources is the same, so no source would be more prominent than the other.

2. Applying the five different PAE methods to extract the primary and ambient sources.

3. Use the extracted outputs and the original sources to evaluate each method using BSS Eval.

4. A baseline is defined by comparing the original ambient or primary sources to the mixture without any separation. This is used to define the improvement of each extraction method over the original mixture.

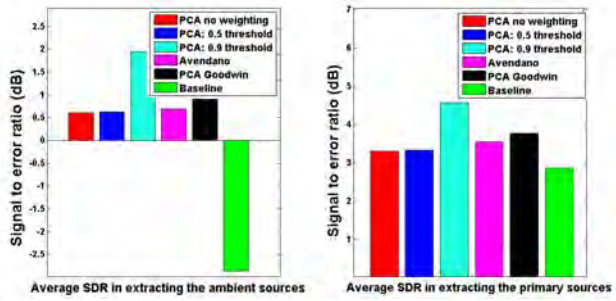


Figure 6: Average SDR in primary and ambient extraction

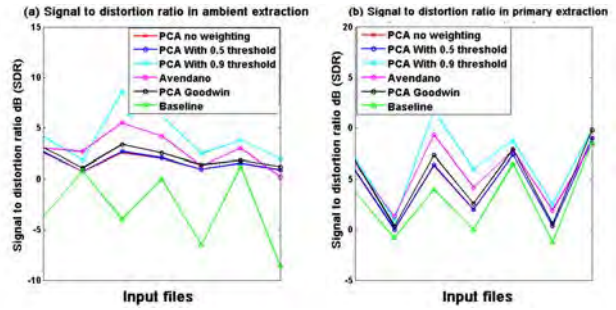


Figure 7: SDR values for a sample of five mixtures

Figure 6 shows the average Signal to Distortion ratio (SDR) in extracting both the primary and the ambient sources for different methods. By analyzing the graph, we find that the proposed weighting shows an improvement in the separation over the other methods. We find that using a higher threshold of 0.9 gives much better separation than using a lower threshold or no threshold. This shows how the weighting improves the accuracy of extraction over both the original PCA and the weighted PCA introduced by Goodwin in [19].

Figure 7 shows the exact SDR values of a sample of five mixtures with comparison to the baseline in both the primary and ambient extraction. We find that all the methods improve clearly over the baseline without separation. In general the SDR values for the primary extraction is higher than the ambient because the primary sources tend to have higher energy.

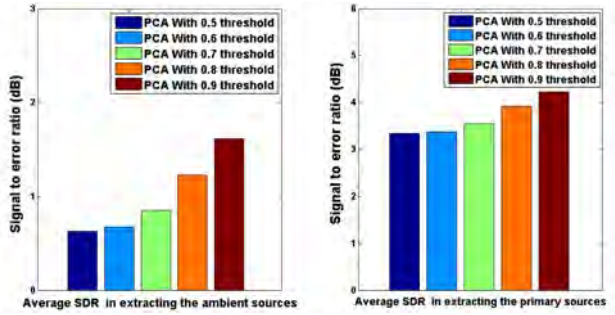


Figure 8: Average SDR for different thresholds

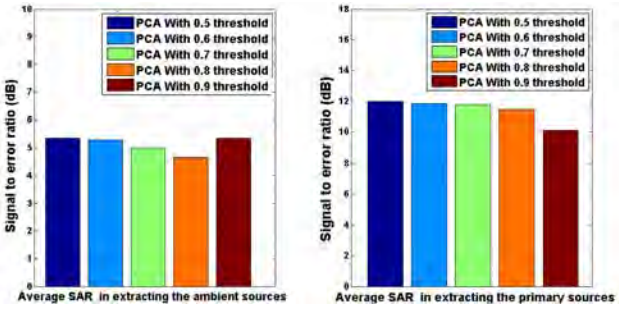


Figure 9: Average SAR for different thresholds

Figure 8 shows how using different thresholds affects the extraction quality. Using higher thresholds gives higher accuracy in the separation, however, extreme weights result in higher distortion caused by the artifacts in the separation process, especially in the extracted primary sources, as shown in Figure 9. Hence, there is a trade-off between sharp separation and artifact distortion. Typically, a threshold in the range $\theta \in [0.6, 0.8]$ would give a proper trade-off between separation quality and artifact distortion.

5. CONCLUSIONS

Separating the primary and ambient sources from an audio mixture shows potential for applications including upmixing an audio recording. In this paper, we explained the need for this separation technique and proper ways of using it in upmixing techniques. We presented a method of extracting the sources using an adaptive Principal Component Analysis (PCA) to solve the common problem of the dominant primary source. The adaptive weighting tests the level of presence of primary sources and ensures to give a proportional weight to both of the sources based on this estimate. The method shows higher separation quality compared to the classic PCA-based separation methods and other methods from the literature. However, this method still shows correlation between the two ambient components leaving room for further improvement in future work. Future work could also include a subjective

evaluation by performing listening test with the different separation methods to ensure the user's experience coincide with the results of the objective evaluation.

6. REFERENCES

[1] V. Pulkki, "Directional audio coding in spatial sound reproduction and stereo upmixing," in *Proc. Int. Conf. Audio Engineering Society: 28th International Conference: The Future of Audio Technology–Surround and Beyond*. Audio Engineering Society, 2006.

[2] M. R. Bai and G.-Y. Shih, "Upmixing and downmixing two-channel stereo audio for consumer electronics," *J. Consumer Electronics*, vol. 53, no. 3, pp. 1011–1019, 2007.

[3] D. Fitzgerald, "Upmixing from mono-a source separation approach," in *Proc. Int. Conf. Digital Signal Processing (DSP), 2011*. IEEE, 2011, pp. 1–7.

[4] B. Xie, "Signal mixing for a 5.1-channel surround sound system'analysis and experiment," *J. Audio Engineering Society*, vol. 49, no. 4, pp. 263–274, 2001.

[5] M. M. Goodwin and J.-M. Jot, "Primary-ambient signal decomposition and vector-based localization for spatial audio coding and enhancement," in *Proc. Int. Conf. Acoustics, Speech and Signal Processing (ICASSP)*, vol. 1. IEEE, 2007, pp. I–9.

[6] M. M. Goodwin, "Geometric signal decompositions for spatial audio enhancement," in *Proc. Int. Conf. Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2008, pp. 409–412.

[7] C. Faller, "Multiple-loudspeaker playback of stereo signals," *J. Audio Engineering Society*, vol. 54, no. 11, pp. 1051–1064, 2006.

[8] C. Avendano and J.-M. Jot, "A frequency-domain approach to multichannel upmix," *J. Audio Engineering Society*, vol. 52, no. 7/8, pp. 740–749, 2004.

[9] S. Kraft and U. Zölzer, "Stereo signal separation and upmixing by mid-side decomposition in the frequency-domain," in *Proc. of the 18th Int. Conference on Digital Audio Effects (DAFx-15)*, 2015.

[10] M. R. Bai and G.-Y. Shih, "Upmixing and downmixing two-channel stereo audio for consumer electronics," *J. Consumer Electronics*, vol. 53, no. 3, pp. 1011–1019, 2007.

[11] C. Faller and J. Breebaart, "Binaural reproduction of stereo signals using upmixing and diffuse rendering," in *Audio Engineering Society Convention 131*. Audio Engineering Society, 2011.

[12] J. Breebaart and E. Schuijers, "Phantom materialization: A novel method to enhance stereo audio reproduction on headphones," *J. Audio, Speech, and Language Processing*, vol. 16, no. 8, pp. 1503–1511, 2008.

[13] W.-S. Gan, E.-L. Tan, and S. M. Kuo, "Audio projection," *J. Signal Processing Magazine, IEEE*, vol. 28, no. 1, pp. 43–57, 2011.

[14] J. Blauert, *Spatial hearing: the psychophysics of human sound localization*. MIT press, 1997.

[15] J. He, E.-L. Tan, and W.-S. Gan, "Time-shifted principal component analysis based cue extraction for stereo audio signals," in *Proc. Int. Conf. Acoustics, Speech and Signal Processing (ICASSP), 2013*. IEEE, 2013, pp. 266–270.

[16] S.-W. Jeon, D. Hyun, J. Seo, Y.-C. Park, and D.-H. Youn, "Enhancement of principal to ambient energy ratio for pca-based parametric audio coding," in *Proc. Int. Conf. Acoustics Speech and Signal Processing (ICASSP)*. IEEE, 2010, pp. 385–388.

[17] J. Merimaa, M. M. Goodwin, and J.-M. Jot, "Correlation-based ambience extraction from stereo recordings," in *Audio Engineering Society Convention 123*. Audio Engineering Society, 2007.

[18] S. Dong, R. Hu, W. Tu, X. Zheng, J. Jiang, and S. Wang, "Enhanced principal component using polar coordinate pca for stereo audio coding," in *Proc. Int. Conf. Multimedia and Expo (ICME)*. IEEE, 2012, pp. 628–633.

[19] M. M. Goodwin, "Adaptive primary-ambient decomposition of audio signals," Jun. 19 2012, uS Patent 8,204,237.

[20] E. Vincent, R. Gribonval, and C. Févotte, "Performance measurement in blind audio source separation," *J. Audio, Speech, and Language Processing*, vol. 14, no. 4, pp. 1462–1469, 2006.

[21] C. Févotte, R. Gribonval, and E. Vincent, "Bss_eval toolbox user guide–revision 2.0," 2005.

A VIRTUAL ACOUSMONIUM FOR TRANSPARENT SPEAKER SYSTEMS

Elliot Kermit-Canfield
Center for Computer Research in Music and Acoustics, Stanford University
kermit@ccrma.stanford.edu

ABSTRACT

An acousmonium, or loudspeaker orchestra, is a system of spatially-separated loudspeakers designed for diffusing electroacoustic music. The speakers in such a system are chosen based on their sonic properties and placed in space with the intention of imparting spatial and timbral effects on the music played through them. Acousmonia are in fact musical instruments that composers and sound artists use in concerts to perform otherwise static tape pieces. Unfortunately, acousmonia are large systems that are challenging to maintain, upgrade, transport, and reconfigure. Additionally, their sole task is limited to the diffusion of acousmatic music. On the other hand, most computer music centers have incorporated multichannel sound systems into their studio and concert setups. In this paper, we propose a virtual acousmonium that decouples an arbitrary arrangement of virtual, colored speakers from a transparent speaker system that the acousmonium is projected through. Using ambisonics and an appropriate decoder, we can realize the virtual acousmonium on almost any speaker system. Our software automatically generates a GUI for metering and OSC/MIDI responders for control, making the system portable, configurable, and simple to use.

1. INTRODUCTION

An acousmonium is traditionally a system with multiple, characteristically sounding speaker groups positioned in space. A performer mixes a stereo audio piece through the system in real-time, imparting spatial and sonic coloration effects on the music as it is diffused. Acousmatic music is specifically written to be rendered through these speaker systems. Composers will deliberately mix their electroacoustic works to be two channels so that there is still the possibility of "interpreting" the music through an acousmonium—the acousmonium is in fact the instrument on which a composer or sound mixer breathes life into a live performance of a fixed media work.

Acousmonia have one purpose: to diffuse sounds through their highly colored speakers. Although they contain many loudspeakers, each speaker's frequency response might be different. Other systems, which we will refer to as transparent speaker systems, contain two ore more loudspeakers that are meant to amplify electroacoustic signals without imparting any timbral change on the signals played through them. That is to say signals passed through these speakers should contain no audible distortion or filtration effects, and should simple amplify the sounds. [1] While an acousmonium can not be repurposed to project amplified audio signals without coloration, a transparent speaker system can render audio for a multitude of applications. In fact, one can use digital signal processing to mimic the colored response of an acousmonium.

Acousmonia hold an important status in computer music history. Even though the concept of the acousmonium is quite old and there have been many technological advances in spatial audio and signal processing, acousmonia are still relevant today. Composers and sound artists still write music to be performed on acousmonia and hold strong opinions on the entire process. An acousmonium has immense creative possibilities for diffusing music concréte and we would be at a great loss without these systems.

Unfortunately, acousmonia are usually large and complex systems that are generally location specific and not particularly portable. Furthermore, one cannot depend on finding an acousmonium in a concert venue. These days, transparent speaker systems for spatial audio are almost guaranteed in concert venues, studios, and computer music centers. For these reasons, we need a way to properly diffuse acousmatic music on a more general, transparent speaker system.

In this paper, we describe a software solution for a portable and reconfigurable virtual acousmonium. Our proposed system allows a user to configure a collection of virtual, colored speakers and place them in space. It automatically generates MIDI and OSC hooks for controlling the system as well as a GUI for metering and audio file playback. By rending the output as an ambisonic encoded signal, this system decouples the virtual acousmonium from the speaker system it is rendered through.

2. ORIGINS

The first acousmonium was designed in 1974 by François Bayle and used by the Groupe de Recherches Musicales (GRM) to diffuse musique concréte through more than 80 loudspeakers [1]. Over time, other acousmonia have appeared, most notably the Gmebaphone and Cybernéphone at the Institut international de musique électroacoustique de Bourges and the Birmingham ElectroAcoustic Sound Theatre (BEAST) at the University of Birmingham [2–4]. All of these systems have been through many reconfigurations and

[1] This is of course impossible.

renovations and are still in use today. Like church organs, these systems are often large and complex. The sheer number of loudspeakers and their associated amplifiers, cables, and mixer channels mean that an acousmonium is typically complicated to setup. Additionally, the acoustics of the space in which an acousmonium is performed is inherently coupled to the timbral and spatial effects of the speakers. For these reasons, acousmonia are often housed in specific locations where the room acoustics and speaker systems compliment one other. [2] Moreover, since these systems are complex, maintenance requires intimate knowledge of the system and they are often built and upgraded over many years. Acousmonia have played an important role in shaping the development of the electroacoustic music tradition, especially in western Europe. For some reason, there seem to be very few such systems outside of Europe. Parallel to the tradition of acousmatic music, electroacoustic music diffusion has also tended towards 3D sound rendering systems. Various techniques have been employed for positioning sound in space (e.g., wave-field synthesis, ambisonics, vector based amplitude panning, etc.). The systems designed for projecting sound using these techniques are highly varied. Simple systems have speakers in a single pantophonic ring around the listeners while more complex systems involve speakers that have height displacement as well. Larger number of speakers in these systems roughly translates to more sophisticated spatial processing and a higher ability to localize sounds in space.

Acousmonium means speaker orchestra. Like a conventional orchestra, groups of speakers have characteristic tonal qualities, spatial locations, and radiation patterns. An artist diffuses a stereo, concréte electroacoustic work through the system much the way a simple melody could be adapted to make use of the full sonic and creative capacity of an orchestra. An acousmonium system contains carefully tailored groups of speakers that have specific purposes for coloring or effecting the sound passed through them. Speakers are characterized by their roles, and include ensemble speakers that produce band-filtered outputs in different frequency ranges, highly colored solo speaker instruments, and effects speakers (e.g., ones for spatial panning, or extreme vertical displacement) [5, 6]. The music is performed live, often from a notated score, and unites a stereo music concréte composition with the interpretation through the speaker system and room. This last step is crucial—composers feel that a composition with more than two channels already has a spatial aspect and that there is no more room for interpretation.

3. JUSTIFICATION FOR A VIRTUAL ACOUSMONIUM

Acousmatic music has a deep musical tradition, but it also has severe limitations. The fact is that these systems are so complex that reconfiguration and transportation become arduous tasks. Maintenance requires detailed knowledge of the system to diagnose problems and upgrade components. Moreover, acousmonia are usually analog instruments that make use of custom mixers, equipment, and electrical components acquired and built for specific systems. Not only does this complicate the upkeep of an acousmonium, but also means that that knowledge does not necessarily translate from one acousmonium to another.

A virtual acousmonium is not bound to a specific space or hardware setup. Our system is designed in a way that it is agnostic to the final diffusion system and can be reconfigured virtually on the fly. Last, this system transcends the traditional acousmonium paradigm. The first goal is to replicate the behavior of an acousmonium—in this case the ability to diffuse a stereo audio piece to a system of virtual, timbrally-colored and spatially separated speakers. We also expand the capabilities of the system so that one can diffuse multichannel works through the system and create hybrid systems of colored and transparent speaker systems.

Our system outputs ambisonic encoded signals allowing us to insert our virtual acousmonium into the signal path of any transparent speaker system. This means that not only can we diffuse acousmatic music in the same concert as music meant to be played through a transparent system, but we can also reconfigure the acousmonium—changing its speaker configuration—between musical works. More importantly, we can reproduce *specific* acousmonium setups, meaning that we can virtually reproduce existing and historic acousmonium systems. This is important for archival purposes and allows us to revive musical compositions that could not be correctly performed due the lack of an appropriate system. It also untethers music written for a specific acousmonium allowing the music to be performed anywhere.

Acousmonia are not simply speaker systems—they are musical instruments in their own right that require years of training to master. Because they are not portable and live in specific spaces, one can not necessarily access the acousmonium in order to practice or experiment with the diffusion of one's music. With our virtual acousmonium, one can approximate specific real-world acousmonia over headphones in order to practice the instrument. Naturally this does not replace working with the true system, although it can serve as a proxy much like a practice organ stands in for the one in the recital hall.

While one can easily up-mix an audio file to a large, transparent speaker system, the same is not true for an acousmonium despite the large number of speakers. The characteristic sound of the speakers renders the these systems useless for applications where the system is not supposed to impart coloration on the signal. At the same time, 3D sound systems have become prominent in computer music institutions. Octaphonic and larger systems, often with speakers displaced at various elevations, are almost guaranteed for concert presentations. Since the virtual acousmonium autogenerates its routing matrix and MIDI/OSC responders, the entire system can be treated as a plug-and-play module that is inserted into a larger system.

Because of the wide availability of transparent speaker arrays, we implemented a framework for performing acousmatic music, in the traditional sense, without a physical acousmonium.

[2] Although some institutions have the means to transport smaller systems for festivals.

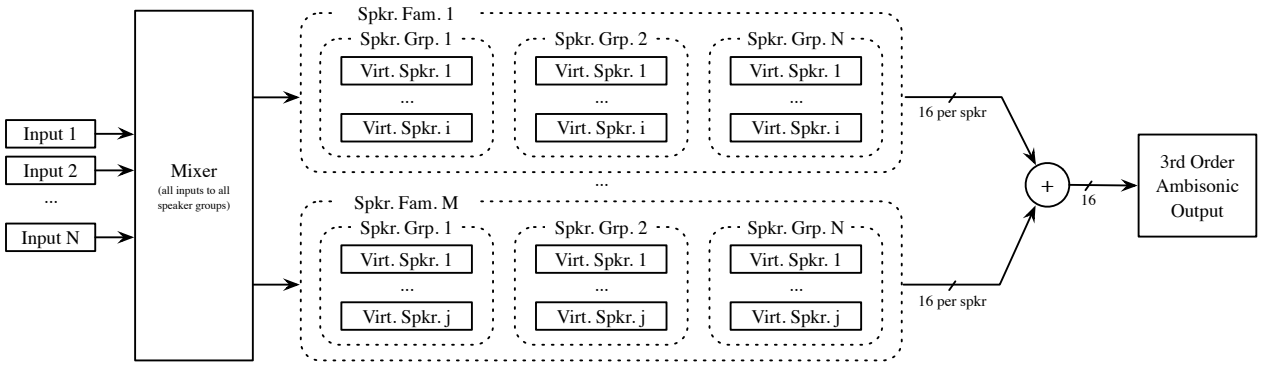


Figure 1. Acousmonium system diagram.

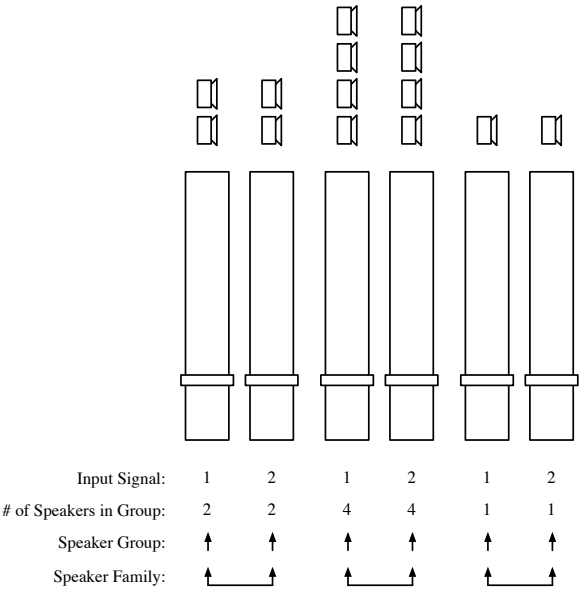


Figure 2. Auto-generated fader arrangement for two source channels and three speaker groups.

4. THE VIRTUAL ACOUSMONIUM

At its core, the virtual acousmonium is a configurable diffusion tool that sits between the audio input and the reproduction system. It is written in SuperCollider to be cross-platform compatible and work with an arbitrary speaker arrangement. We provide a simple configuration for interfacing the software with audio and MIDI hardware and default/example configurations for virtual speaker descriptions. Our virtual acousmonium provides convenient methods to load and playback audio files as well as a mechanism for reading input directly from the sound card. When you load a speaker configuration, the system auto-generates the appropriate virtual speakers, audio routing matrix, OSC/MIDI wrappers for controlling the system, and a GUI for monitoring the system.

4.1 Software

We wrote this virtual acousmonium in SuperCollider because it provides the right combination of efficient signal processing and configurability. SuperCollider is an open source audio programming language and environment with a strong developer community and user base. Additionally, SuperCollider handles multi-channel audio data in a convenient way for implementing complex audio routing.

In our benchmarks, running a complex system with 2 input channels and 80 virtual sources (each with its own ambisonic panner), the peak cpu load was less than $\frac{1}{4}$ of a single core of a 2011-era i7 processor, and the average cpu load close to 10%. This tool free and open source software distributed under a GNU GPL. [3]

4.2 Speaker Descriptions and Routing

Virtual speakers in our system are described by SuperCollider SynthDefs that accept a monophonic input and output 3^{rd} order ambisonic encoded signals. Each virtual speaker has an associated angle (θ), azimuth (ϕ), and relative gain (γ) that describe where in space the virtual speaker should appear. We use open source SuperCollider plugins for higher-order ambisonic encoding.

Since the speakers are defined in SuperCollider code, custom speaker responses can be achieved by extending the speaker definitions we have included. As long as a speaker definition respects our speaker definition format, arbitrary code can be evaluated to model the desired speaker responses. This processing can take the form of simple signal processing algorithms like filtering, distortion, or compression, but could also include a full physical model of a specific speaker, horn, or driver. The format for the speaker definition includes several reserved arguments (e.g., spkrAzmth, spkrGain, etc.), a method for sending OSC messages to update meters, and a unit generator graph that conforms to a "monophonic input $\rightarrow$ processing $\rightarrow$ ambisonic panner" paradigm. In addition to the reserved arguments, the system has appropriate mechanisms for interpreting extra parameters that are not generalized for all

[3] This virtual acousmonium software can be downloaded at `https://ccrma.stanford.edu/~kermit/website/acousmonium.html`.

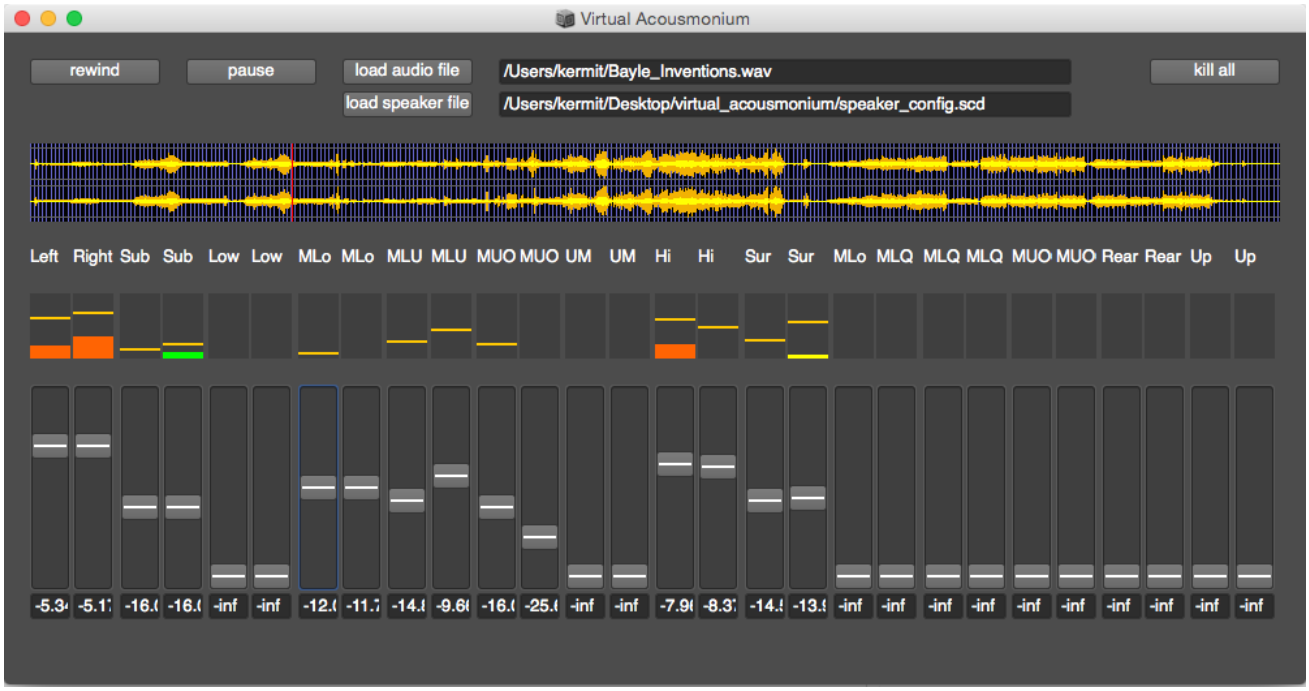


Figure 3. SuperCollider QT graphic interface.

speaker descriptions (e.g., a speaker with a filter might need a mechanism for setting the filter cutoff frequency and Q value).

The speaker routing in our system is described by a hierarchical configuration file. Virtual speakers can be grouped together to create the impression of more complex speaker arrangements, and multiple groups of speakers can be collected into "families" in order to facilitate control over multiple, similar-but-spatially-separated speaker sets. A full system block diagram can be seen in Fig. 1. The speaker configuration is stored as an array of speaker families, where a speaker family is an array of speakers and each speaker is a dictionary of associated parameter keys and values, see code snippets 1 and 2.

Code 1. Data structure for the speaker configuration

```
[          // spkr family containing
  [(spkr), (spkr), ...], // spkr grps
  [(spkr), (spkr), ...],
  ...
]
```

Code 2. Single virtual speaker description

```
(
  \spkrType  : \speaker_hp,
  \name      : \Mid_Range_L,
  \params    : (
    \spkrAzmth : (-pi/4),
    \spkrElev  : 0.0,
    \spkrGain  : 1.0,
    \cutoff    : 2000,
    \rq        : 1.0
  )
)
```

The output of all the virtual speakers gets summed together and sent out of the system effectively decoupling the virtual speakers from the physical, real-world speaker system. Since the system description is just a piece of software, they speaker arrangement is flexible and can be reconfigured on-the-fly to accommodate compositions that are written to be diffused through specific speaker setups.

4.3 Interfaces

From the speaker configuration, SuperCollider will automatically generate a QT GUI and set of OSC and MIDI responders for the system. The system provides both an interface for processing external inputs to the SuperCollider program as well as a way to load an audio file into an internal playback system. If the later is used, basic transport controls (play/pause, seek, rewind) and a waveform plot with playhead are provided for navigating the sound file. [4]

Inherently, all inputs are mapped to all speaker groups. The number of speaker groups determines the number of fader groups and the number of inputs the actual number of faders, see Fig. 2. This is then exposed to the users as a GUI fader bank. Each fader can be controlled with OSC or MIDI allowing the use of external interfaces. The system automatically generates faders and level meters as well as OSC and MIDI responders for each speaker group, based on the hierarchical speaker description in the config file.

An example of the GUI generated for a stereo input signal and 24 virtual speaker groups is shown in Fig. 3.

4.4 Limitations

One major issue with using virtual sources placed in space with ambisonics instead of real speakers is the fact that

[4] Very soon the framework will also support computing and displaying a spectrogram view of the sound file as well.

it is challenging to accurately simulate speaker radiation patterns. Physical speaker enclosures have nonlinear radiation patterns that are impossible to reproduce in a virtual ambisonic environment. Even so, we can use reverb, and filters, as well as a cluster of point sources with anti-phase components to approximate speaker orientation, depth, and radiation patterns.

5. CONCLUSIONS AND FURTHER WORK

In this paper, we have introduced a tool for generating arbitrarily complex arrangements of spatially-located and colored virtual speakers and interfaces for diffusing audio through the system. By encoding the output of this virtual acousmonium using ambisonics, our system is modular and can be used with any transparent speaker system. Our system automatically generates a graphic user interface for interacting with and monitoring the system as well as providing OSC and MIDI responders so one can control the system with external devices.

In the future, we intend to add a graphical interface for creating the speaker configuration files. While convenient for programmers, the current textual method does not provide any visual feedback that speakers are placed in the correct locations. We envision a tool where a system designer can enter speaker locations and types in a context where there is no ambiguity of the virtual speaker's positions.

This tool does not seek to replace acousmonia, but rather complement these impressive systems. The portability and configurability of a virtual system increases the umbra of this musical tradition, and the signal processing possibilities of this tool will benefit composers and diffusion artists alike.

6. REFERENCES

[1] F. Bayle, *Musique acousmatique: propositions... ...positions*. Buchet/Chastel, 1993.

[2] C. Clozier, *Proceedings volume III of the 1997 works of the International Academy of Electroacoustic Music*. Editions Mnemosyne, 1998, ch. Composition/Diffusion in Electroacoustic Music, pp. 233–281.

[3] J. Harrison, "Diffusion: theories and practices, with particular reference to the beast system," *eContact!*, 1999.

[4] H. Tutschku, "On the interpretation of multi-channel electroacoustic works on loudspeaker orchestras: Some thoughts on the grm acousmonium and beast," http://www.tutschku.com/content/interpretation.en.php.

[5] J. Prager, "L'interpretation acoumatique: fondements artistiques et techniques de linterpretation des œuvres acousmatiques en concert," 2002.

[6] A. V. Gorne, "L'interprétation spatiale. essai de formalisation méthodologique," *Revue Déméter*, December 2002.

POLYTEMPO COMPOSER: A TOOL FOR THE COMPUTATION OF SYNCHRONISABLE TEMPO PROGRESSIONS

Philippe Kocher
Institute for Computer Music and Sound Technology
Zurich University of the Arts
philippe.kocher@zhdk.ch

ABSTRACT

The accurate synchronisation of tempo progressions is a compositional challenge. This paper describes the development of a method based on Bézier curves that facilitates the construction of musical tempo polyphonies up to an arbitrary level of complexity, and its implementation in a software tool. The motivation for this work is to enable and encourage composers to create music with different simultaneously varying tempos which otherwise would be too complex to manage.

1. INTRODUCTION

Computers have become an important tool for composers, but very few commercial music applications allow for the generation of multiple simultaneous time streams. This is understandable, as most music of our culture takes place in one single time stream and polytemporal music is quite uncommon. Hence, there is a very small demand for such applications. Composers who nevertheless want to engage in tempo polyphony have to resort to computer music programming environments and implement their own tools, as Dobrian shows exemplarily for the programming environment Max [1]. Yet not every composer is at the same time a skilled programmer and without the availability of appropriate tools, only few composers can create polytemporal music and explore the field of tempo polyphony.

Whereas it is a easy task to construct a succession of musical events that change in tempo just by gradually increasing or decreasing the events' inter-onset intervals, the calculations for tempo progressions that eventually arrive at a defined point in time are not trivial. Nevertheless, these calculations are an essential prerequisite for the synchrony between simultaneous voices in polytemporal music; they are needed, for instance, to construct convergence points of time streams at which musical events of different polyphonic layers converge after a section of independent changes in tempo. Different ways to approach this problem are object of previous studies in the field [2, 3].

This paper proposes a tool to devise temporal polyphonies aimed at composers without in-depth mathematical knowledge. It provides a GUI for the intuitive manipulation of complex tempo structures and the facility to audition the tempo progressions. Another important feature of this tool is its metrical flexibility: It does not restrict the user to regular bar and beat patterns, which is a shortcoming of many similar tools. Tempo polyphonies created with this tool can be exported in different formats to synchronise performing musician or for further use in other computer assisted composition environments.

2. MUSICAL CONTEXT

The early 20th century witnessed the rise of many novel compositional techniques to handle musical time. By employing complex, asymmetrical or 'free' rhythms, most of these techniques were intended to overcome musical metre that had dominated western culture for many centuries. Some composers even attempted to negate the traditional notion of a common unifying tempo at all and divided the music into several layers to be played by different musicians in different tempos. Early examples of such music can be found among the compositions of Charles Ives.

2.1 Music in Need of Technology

Between 1916 and 1919 Henry Cowell wrote his book *New Musical Resources*, in which he unfolded his theories about the relation of rhythms, metres and tempos to the ratios of the overtone series [4]. He admitted that most of the rhythms he described were too complex to be played by a musician, but he suggested that they be cut on a player piano roll. Cowell never used such a self-playing piano himself, but his book inspired Conlon Nancarrow to compose his *Studies for Player Piano,* which up to the present day still constitute the most substantial corpus of polytemporal music. However, this music is entirely mechanical and overcomes the limitations of human performers by avoiding them altogether.

Polytemporal composition that are to be played by human musicians (i. e. conceived for acoustical instrument that have to be played by human musicians) require a performance aid to keep the tempo. Metronomes or click tracks are commonly used in such situations to allow for a timing accuracy beyond the capabilities of a human performer. In a similar vein one could argue that composers also might benefit from a mathematical aid to devise intricate tempo relationships and tempo progressions whenever the degree of complexity is beyond their mathematical capacity. In

this sense, the focus is put on the composer's need of technological aids in the remainder of this paper.

2.2 Compositional Approaches

There exist different compositional approaches to handle polyphonic tempo structures. Many works deal with the superposition of different independent musical layers. No matter whether these layers are to be played freely or marked with a precise metronome indication, the exact coincidence of any musical events cannot be predicted, because, due to the inaccuracy of the human performance, the tempos will always drift apart to a certain extent. It might be argued that this kind of music is better described as 'multi-layered' than truly polyphonic. Needless to say that composers have always been aware of this unavoidable rhythmic blurring and have organised the harmony, melody, gesture etc. of their music accordingly.

Another compositional approach consists in creating a polytemporal music that does not only rely on the expressive quality of stratification *per se,* but is based on a tight synchronisation of all musical events as, for instance, the aforementioned *Studies* by Nancarrow. It must be pointed out that such a temporal precision is not only an issue of rhythm, but also a subject of harmony, as harmony itself is based on the synchronicity of pitched musical events.

What is required to compute the accurate timing of all musical events in a polytemporal counterpoint? The superposition of different tempos (especially in simple ratios) is trivial, but the superposition of different tempo *progressions* (i. e. accelerandos or decelerandos) is a difficult task. The difficulties lie not only in the intricacies of the mathematic formulas, but also in the not entirely obvious and intuitive interdependence of the parameters involved, such as elapsed time, note value, tempo and a specific change in tempo (given as a characteristically shaped tempo curve).

3. DEFINITIONS

3.1 Tempo

One of the first things a novice musician must learn is a sense for a regular pulse. The ability to accurately perform a (notated) rhythm, no matter how complicated or asymmetrical it might be, is based on this skill. This notion of a regular pulse is deeply ingrained in our musical practice, thus we know the concept of *beat* and assume that it is a constant unit of time. Hence, *tempo* is usually defined as rate of the pulse or beat, measured in beats per minute. This definition, however, is problematic for our purposes for two reasons: First, the unit of the beat is not explicitly defined, which would be necessary in order for the tempo to be handled mathematically. Second, even though the concept of a constant unit of time is a basic principle of music making, this definition rules out metrical structures with non-isochronous pulses, i. e. asymmetrical metres or changes between simple and compound metres. A more open definition of tempo avoids the reference to a countable beat: tempo is *note value per time unit.* For example, one crotchet per second is tempo $= 0.25s^{-1}$.

3.2 Score Position

The position in the score is often referred to as 'symbolic time' or 'score time', which both are unfortunate terms, as the note values of the symbolic music notation only express duration *ratios,* which only become actual physical durations only when executed in a certain tempo. Therefore, the terms *score position* or *score distance* are preferred. The timing of every sonic event is defined by a location in time and a position in the score. Just as time is continuously incremented, the source position can be measured from an established zero-point as well, e. g. the second crotchet in the tenth $^4/_4$ bar is at a score position of $9^1/_4$ etc.

3.3 Tempo Map and Time Map

There are two graphical representations of musical time: the *tempo map* and the *time map*. The tempo map depicts speed as function of time, the time map score position as function of time (see Fig. 1). In either representation one parameter always remains implicit: In the tempo map the score position is the integral of the curve, in the time map the tempo is the derivative of the curve.

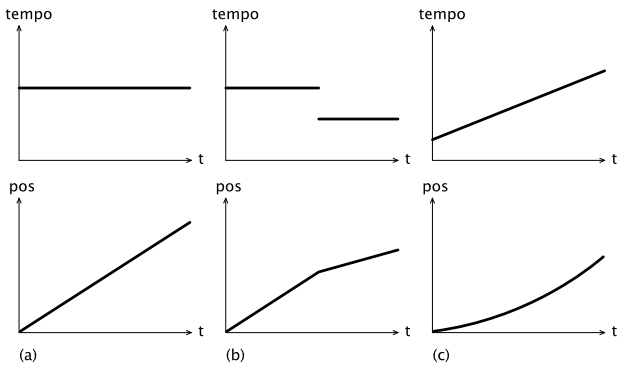


Figure 1. Three tempo maps and their equivalent representation as time maps; (a) depicts a constant tempo, (b) a sudden tempo change, (c) an accelerando.

The tempo map seems to be the most natural way to describe a tempo progression. It is also the starting point for most methods to calculate varying tempos. The time map was introduced by Jaffe [5] as means to realise expressive timing while maintaining overall synchronisation. It is mainly used in research into performance practice to describe timing deviations or rubato, as function to associate 'idealised' time to 'actual' time [6–8].

The objective of the method presented in this paper is not the formalisation of expressive devices such as rubato, but the synchronisation of individual tempo progressions. As soon as it comes to synchronisation, all calculations based on the tempo curve face a intricate problem: The alignment of two musical events (e. g. two downbeats) in two different independent tempo streams does nearly always require a certain amount 'error correction', i. e. a function that distorts the tempo curve appropriately. When using a time map, on the other hand, the synchronisation of events is obvious and as easy as the horizontal alignment of points.

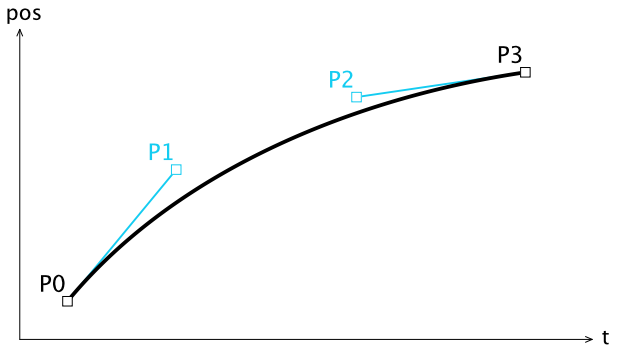


Figure 2. A cubic Bézier curve defined by four points. If drawn on a time map, P_0 and P_3 represent assignments of score locations to exact times; P_1 and P_2 control the shape of the curve, especially the slope of the curve at P_0 and P_3 which describes the initial and final tempo.

4. MATHEMATICS

For the purpose of synchronisation, it is necessary to assign exact times to certain score positions. These assignments are called *control points* and are set in the time map to build the temporal scaffold of the music on which the timing of all musical events depends. Once these control points are set, they are connected by Bézier curves (a solution already proposed by Berndt [8]). As shown in Fig. 2 a cubic Bézier curve is defined by four points P_0, P_1, P_2 and P_3. The curve starts at P_0 and ends at P_3. The additional points P_1 or P_2 provide directional information and determine the shape of the curve. Their relative position to P_0 and P_3 determine the slope at the beginning and the end of the curve, which in our case represents the initial and final tempo of the section. The distance between P_0 and P_1 or between P_2 and P_3 is used to weight the respective tempo, i. e. to determine how long the initial tempo is held or how soon the final tempo is reached. The time curve (and hence implicitly also the tempo curve) can be warped by moving these two additional points. This can be used, within reasonable limits, to adjust the shape of the tempo progression in order to achieve the desired gestural quality.

5. IMPLEMENTATION

In order to make it readily accessible for composers, this formalism has been implemented in a standalone application. The application is programmed in C++ using the framework JUCE [1] . The user interface is structured as follows: On the topmost level, the temporal polyphony is organised in sequences. Every sequence represents one concurrent tempo stream, i. e. a polyphonic layer or the part of an instrument in a ensemble, when described in terms of a traditional score. Each sequence consists of an event pattern and a list of control points. The event pattern describes the sequence's rhythmical structure and the control points match time and score position (see Fig. 3).

[1] `http://www.juce.com` (URL valid in July 2016)

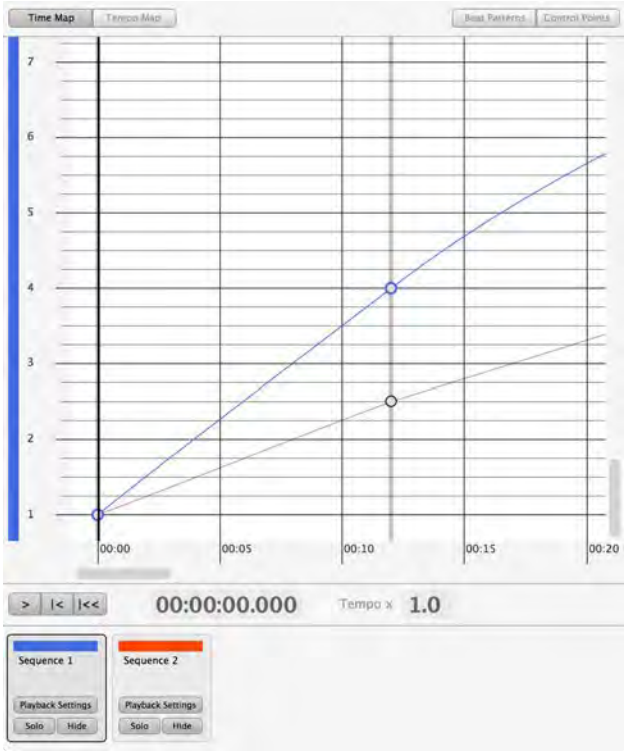


Figure 3. A screenshot of *Polytempo Composer:* two tempo streams (sequences) depicted on a time map. Both streams are in different tempos (expressed by the steepness of the curve) and synchronised at one point around 00:12 seconds.

$\frac{4}{4}$	$\frac{3+2+2}{8}$	$\frac{2}{4}+\frac{1}{16}$
4/4	3+2+2/8 *or* 3/8+2/8+2/8	1/4+1/4+1/16 *or* 4+4+1/16

Figure 4. The input format of the event pattern is oriented towards the traditional music notation and allows for complex and additive metres.

5.1 The Event Pattern

The event pattern is expressed in score position units. In case of a traditionally conceptualised and notated music, these units can, for the sake of simplicity, be equaled with rhythmical values. If one intends to generate a metronome track, the event pattern is set to correspond to the metrical structure of the music, i. e. every event represents one beat. For convenience, the metres can be entered in the usual way as fraction, whereby a '+' is used to define non-isochronous metres (see Fig. 4). If the music is based on varying metres, the event pattern is composed of several sub-patterns, each of which represents one bar an can be repeated multiple times (see Fig. 5).

If the intended output is not a metronome track, the event pattern does not have to represent a metre. Rather, it can

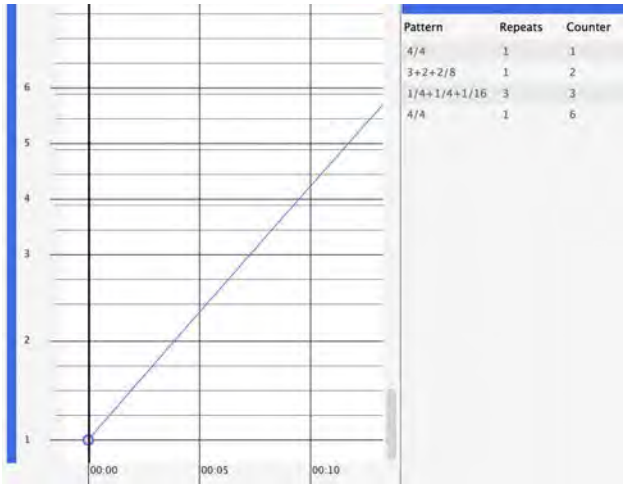


Figure 5. To generate a metronome track, the event pattern corresponds to the bar/beat structure of the music. A list of sub-patterns (on the right) constitute the overall event pattern. The metrical structure is reflected in the marks on the vertical axis of the time map (on the left).

Time	Position	Tempo In	Tempo Out
0	0	—	0.25
10	3	0.25	0.25
20	5	0.25	0.25
40	10	0.25	—

Figure 6. List of control points specified by time, position, incoming and outgoing tempo. The tempos are indicated in units per second.

be put together as a list of proportional units to form any arbitrary, periodic or irregular, rhythmic structure.

5.2 Control Points

A control point is a precisely defined assignment of a certain score position to a certain time. It can be input graphically directly on the time map or numerically in a table (see Fig. 6). Apart from time and score position each control point is further defined by an incoming and an outgoing tempo whose values determine the slope of the curve at this point. Usually these two tempos are the same. However, in order to express a discontinuity in tempo, e. g. a sudden change of speed or a tempo modulation between two proportional tempos, two different values have to be input. Tempos are specified in units per second. Users who find this counterintuitive can change the tempo format to the familiar 'crotchets per minute' in the global settings.

Helper functions are provided to move a control point automatically to a location where it meets certain criteria: A point can be placed, for instance, in such a way that a constant tempo is kept from the previous point on.

5.3 Output

The exact time of every event is calculated from the tempo curve that is generated as cubic Bézier curve interpolation between the control points. The events timed in such a way can be output in various ways.

The events can be played back from within the software using differently pitched notes (like a metronome) and, if desired, each sequence on a different hardware audio channel. This audio playback feature is primarily designed for the composer to listen to any tempo polyphony under construction, which is essential, as composers need their ears, rather than their eyes, to judge if a tempo progression does work. Likewise, this feature can be used to generate a click track in real time. Furthermore, every event can be assigned a MIDI or a OSC message, which opens up possibilities for various live performance settings.

For further use in any computer assisted composition environment, a list of all calculated times can be exported as plain text (optionally comma-separated or enclosed in brackets). In case the events represent the metrical structure of the music, the data can be written to a JSON file specifically formatted to be seamlessly transferred to the associated virtual conducting software *PolytempoNetwork* [9].

6. DISCUSSION

Synchronising tempo progressions is a difficult task. But it is a prerequisite to control temporal polyphonies that consists of more than a stratification of only loosely coordinated layers of music. It is is fact unimaginable that composers conceive a tightly controlled tempo polyphony without the aid of a mathematical tool. Moreover, such a tool can, presumably, help the composer to learn about the fundamental mathematical principles of tempo progressions, and to acquire an important experience for the conception of more advanced polytemporal music. Tempo progressions or tempo curves are defined by several characteristics: initial and final tempo, elapsed time, covered score distance and shape of the curve. All these characteristics are interdependent, wich the composer can explore through interaction with the tool.

For example, one might decide that a tempo change does not sound right, because an accelerando is, say, 'too early too fast'. In such a case, changing the shape of the curve is not the only measure to take (and often does even not yield the desired results). Rather, one could also alter the initial or final tempo, drag one of the control points to another vertical or horizontal position (i. e. change its time or score position) or do a combination of all these.

The interdependence of parameters also appears very clearly when one attempts to produce an 'impossible' tempo progression. When, for example, the tempo is too fast for a too short a distance in the score, the time curve becomes automatically s-shaped to compensate the timing error (see Fig. 7). The resulting tempo fluctuation depends on the amount of distortion of the curve; it may range from an almost imperceptible rubato to a strong momentary change of speed.

Finally, one point of critique has to be mentioned. As the presented method operates on the time map, it allows for control over the time curve but does provide no (or only implicit) control over the tempo curve. This might be un-

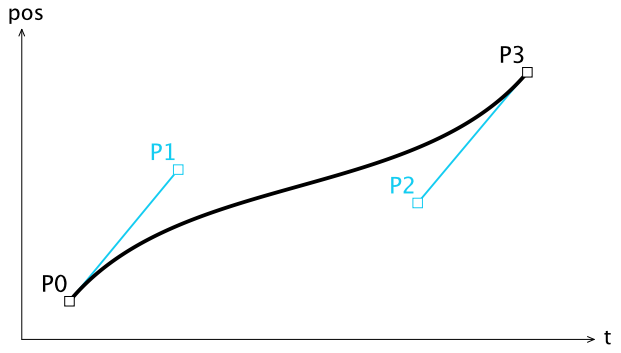


Figure 7. The tempo at P_0 and P_3 is the same. However, this tempo is too fast for too short a score distance (or too long a duration). To compensate for this error, the time curve becomes s-shaped.

usual or even unwanted for some composers. However, due to the fact that synchronisation is the first and foremost feature of this method, this seems to be a sensible compromise.

7. CONCLUSION & OUTLOOK

This paper described a method to compute synchronisable tempo progressions and its implementation in a software tool that can potentially aid the composer in realising polytemporal music with complex tempo relationships. The presented implementation is sufficiently general to be used in the context of symbolically notated instrumental music as well as electroacoustic music. It facilitates the exploration of complex tempo structures even for composers who do not want do engage with the intricacies of the underlying mathematics.

It is hoped, that composers will make use of this software and thus create different user scenarios that will stimulate the further development and improvement of this software. Apart from being a tool for experimental polytemporal music, this software could also be a useful tool for film music, when ever it is needed to synchronise specific points in the score tightly with the film.

Future developments might also include score typesetting. No existing music engraving software of high quality provides a straightforward facility to notate different tempos at the same time, let alone different varying tempos. A notation tool that allows for the notation of different parts of a score in different tempos would be beneficial for composers of instrumental music.

The software *Polytempo Composer* and its source is freely available for download from the project's website: `http://polytempo.zhdk.ch` (URL valid in July 2016).

8. REFERENCES

[1] C. Dobrian, "Techniques for Polytemporal Composition," in *Proceedings of Korean Electro-Acoustic Music Society's 2012 Annual Conference*, Seoul, 2012.

[2] J. MacCallum and A. Schmeder, "Timewarp: A Graphical Tool For The Control Of Polyphonic Smoothly Varying Tempos," in *Proceedings of the International Computer Music Conference*, 2010, pp. 373–376.

[3] J. C. Schacher and M. Neukom, "Where's the Beat? Tools for Dynamic Tempo Calculations," in *Proceedings of the International Computer Music Conference*, 2007, pp. 17–20.

[4] H. Cowell, *New Musical Resources*, D. Nicholls, Ed. Cambridge: Cambridge University Press, 1996.

[5] D. Jaffe, "Ensemble Timing in Computer Music," *Computer Music Journal*, vol. 9, no. 4, pp. 38–48, 1985.

[6] P. Desain and H. Honing, "Tempo curves considered harmful," *Contemporary Music Review*, vol. 7, no. 2, pp. 123–138, 1993.

[7] H. Honing, "From Time to Time: The Representation of Timing and Tempo," *Computer Music Journal*, vol. 25, no. 3, pp. 50–61, 2001.

[8] A. Berndt, "Musical Tempo Curves," in *Proceedings of the International Computer Music Conference*, Huddersfield, 2011, pp. 118–121.

[9] P. Kocher, "Polytempo Network: A System for Technology-Assisted Conducting," in *Proceedings of the International Computer Music Conference*, Athens, 2014, pp. 532–535.

THIS IS AN IMPORTANT MESSAGE FOR JULIE WADE: EMERGENT PERFORMANCE EVENTS IN AN INTERACTIVE INSTALLATION

Brent Lee
University of Windsor
brentlee@uwindsor.ca

ABSTRACT

This is an important message for Julie Wade exists both as a multimedia performance piece and as an interactive audiovisual installation. Each version of the work was publicly presented between 2013 and 2015; a recent 2016 version seeks to incorporate elements of both of the earlier performance and installation versions. Currently, the work may be installed in a gallery space and controlled by Max/Jitter patches that randomly generate related audio and visual events; at the same time, this new installation remains open to interactive musical performance, responding to such interventions with extra layers of sound and revealing an extra set of video clips. This hybrid environment raises a number of ontological questions. In what ways are these musical interactions with the installation "performances"? Is a musician interacting with the installation a "performer" in the conventional sense, and does this interaction impose a different role on a casual gallery visitor witnessing this interaction? Can this mode of presentation of an audiovisual work transcend the limitations of conventional performance and installation? This paper explores these questions within the context of the evolution and 2016 presentation of *This is an important message for Julie Wade*.

1. INTRODUCTION

This is an important message for Julie Wade is an audiovisual piece created from found sound and dozens of video clips gathered over a period of ten years. The process of developing *This is an important message for Julie Wade* began in 2002 with a series of automated phone messages intended for a woman named Julie Wade who presumably had been associated with the phone number assigned to us when we moved to our new city. The calls invariably began with a male voice saying, "This is an important message for…" followed by another male voice saying "Julie Wade". The message would continue, requesting that Julie Wade contact a certain phone number and quote a certain file number. For many months I simply deleted the messages as they came in, but when the messages starting to vary in subtle ways (different voices, different texts), I began saving them with the idea of eventually using the messages as source material for some sort of sound art piece.

The first version of the work was presented as an audiovisual installation in 2013; ten years of phone messages were edited into multiple tracks of audio, some of which only include the recitation of the phone and file numbers. These audio tracks are randomly accessed through a Max algorithm and dispersed through four channels to loudspeakers surrounding the installation space. Black and white video projected on one wall of the gallery features a noisy and indistinct image occasionally interrupted by short video clips of numbers gathered from road signs, advertisements, and building addresses. An additional background layer of audio was created from heavily processing the sound of the ringing telephone to create a murky and dense sonic texture. The installed work has an interesting effect, as it juxtaposes the cheerful tone of the text recitations, the anxiety related to repeated messages from a collection agency, and the multiple levels of randomness reflected in the origins and the treatment of the audiovisual material.

The following year I developed a performance version of *Julie Wade* for inclusion in a program of multimedia performance pieces. In the performance version, the sound of a live instrumentalist is used to trigger an additional set of video clips featuring a woman sitting in a room, ostensibly Julie Wade herself waiting for a phone call. The musician improvises unusual sounds on their instrument (squeals, thick dissonances, multi-phonics, blurred flurries of notes, etc.); the modified Max algorithm brings up one of the Julie Wade video clips with each new live sonic gesture. The performance version is framed by the playback of contrasting but complete and unaltered phone messages, and has been performed and well-received multiple times.

2. PERFORMANCES, INSTALLATIONS, AND INTERACTIVITY

A performance is, among other things, a unique event in a specific time frame, usually with more or less clear beginning and end points.[1] With most performances, these

[1] An action within a time frame is one of multiple features (contestably) intrinsic to performances across cultures and artistic disciplines. For a more in depth discussion of the nature of performance see Carlson [1].

points are equally understood by the performers and the viewing audience, and are conventionally marked by ritual actions such as the entry of a performer onto a stage or the applause offered by the audience[2] at the end of the performance. Over time these conventions have been challenged in any number of ways, such as beginning a performance surreptitiously or by surprise, requiring voluntary or involuntary audience participation, dispersing performers into space normally reserved for the audience, or presenting very long performances that invite the audience to come and go as they please. Still, the drama and significance of such challenges to convention are often premised on the audience's expectation of the rituals that surround performance events.

Installation works also reflect on some level a similar consideration of a set of conventions and rituals. Anne Ring Peterson suggests that installations share three fundamental characteristics: 1) that they activate "space and context", 2) that they stretch an artistic work in time, and 3) that they focus on the "viewer's bodily and subjective experience" [2]. A performance within an installation by its nature addresses its situation in three-dimensional space as well as the expected behaviour of those experiencing the work. While the physical boundaries of the installation space may be generally understood by gallery visitors, the timeframes within which visitors experience an installation differ widely and are not likely to be coordinated. As with performances, rituals of participation apply, though they tend to be the rituals of the art gallery: visitors navigating the installation for a few moments before proceeding to the next gallery space, or perhaps experiencing the installation in the context of an opening, with a glass of wine and friendly chat. The characterization of performances as distinctly temporal and installations as distinctly spatial is not new,[3] but does carry implications for the way a work is experienced: audience members share an experience in a given time, while gallery visitors share an experience in a given space.[4]

Interactive is a term that is often applied to varieties of both performances and installations. For musicians and dancers, *interactive* usually describes a relationship between the performer and the work, while within installations and other genres rooted in a visual arts tradition, *interactive* describes a relationship between the viewing audience and the work. This distinction becomes important when broadly considering the nature of interaction in interactive audiovisual installations. The use of microphones, cameras, sensors and software to capture data representing sound and movement creates the interactive interface between the work and the individuals in the installation space.[5] But the level of complexity and sophistication of sound and movement generated by trained artists exceeds the level that can be expected of a general public, and thus an interactive installation that requires training and preparation of its "performers" presents another world of artistic possibilities, a world that exists in all of the performing arts.

3. THIS IS AN IMPORTANT MESSAGE FOR JULIE WADE

In what ways are musical interactions with an installation "performances"? Is a musician interacting with the installation a "performer" in the conventional sense, and does this interaction impose a different role on a casual gallery visitor witnessing this interaction? The multiple versions of the *Julie Wade* piece have created an opportunity to isolate these questions and examine them from different perspectives.

In the original installation version of *Julie Wade* there is no performer; a Max algorithm controls the playback of audio and video clips while gallery visitors come and go as they please. This first version isn't interactive at all (perhaps *generative* would be a better work to describe it), though the idea of feeding a signal from a microphone into the Max patch to trigger the extra layer of video and reveal the waiting Julie Wade emerged from the process of working on the early installation version. This idea was implemented in the "performance version" of the piece, where a series of saxophone multi-phonics that blend well with the other sonic elements is used as the interactive trigger. Based on a simple patch that measured amplitude changes to trigger the extra video clips, it was easy to in turn incorporate that element into the installation version with the idea that visitors could trigger the extra video clips with any sound that was loud enough (clapping, finger-snapping, etc.).

Distinctions between the two versions emerged. The use of a musical instrument in itself suggested "performance", even as the piece developed. Finger-snapping was an integral part of testing the patch, but testing the patch with the saxophone gestures felt more like *rehearsing* than *testing*. The possibility of taking turns at the microphone to trigger the video clips exists in the installation version, but the performance version restricts the triggering to a designated instrumentalist. Taking turns confers no special status on anyone interacting with the

[2] I am using the term *audience* for listeners in a concert performance situation, *viewing audience* for those attending a dance or theatre performance, and *gallery visitors* for people experiencing an installation. I use the term *performer* in all situations, even though the term is not quite right for even the hybrid installation context. These choices remain difficult, as the terms connote different levels of passive or active participation, and in themselves confer roles on individuals that the work is in part designed to question. This issue of terminology is central to audience studies, and has been explored by numerous scholars. The 2010 issue of the journal *About Performance* is devoted to audience studies, and several of the essays included therein address the implications of terminology; Laura Ginters' introductory article offers an overview of recent research focusing on audiences, spectators, and their roles [3].

[3] Gascia Ouzounian's dissertation on sound art and spatial installations thoroughly explores this idea, positing different varieties of spaces within which installations can be situated [4].

[4] Of course performances still take place in space and installations are still experienced over time; without oversimplifying, the practices and rituals that inform performances and installations differ in part in their temporal and spatial considerations.

[5] David Saltz distinguishes between interactive computer art in which interactivity is a feature of a performance and art in which the interactivity constitutes an environment. In the former case, the interaction still produces performances that are tokens of a "work", be that work a play, a musical composition, a dance or a ritual. In the latter case, interaction does not produce a token of a work, but is inextricably a part of the work itself [5].

installation, but my rehearsal of the saxophone gestures and ability to create a musical shape over time put me in a position to make a performance in a way that a finger-snapping gallery visitor never could. At the first public performance of the piece in Norway in 2014, the saxophone interface in a sense produced both a performer and an audience: we accepted these roles that the situation had created for us.

Building a performance piece from the raw materials of the installation posed its own challenges. The various audio elements of the installation version became a background for the musical gestures of the live musician(s), and live processing of the audio signal coming from the musician(s) helped create a connection between the acoustic and electronic sound sources. The performance version relies to a large extent on the improvising musician(s) to create a structure in time that manages to reflect the randomness that is part of the character of the piece yet still holds the attention of an audience for several minutes. The performance version is framed by complete and unedited statements of different versions of the original phone messages that function as cues that the performance has begun and ended.

The 2016 hybrid version of *Julie Wade* combines elements of the installation and performance versions in an attempt to overcome some of the limitations of each medium. The hybrid version is realized in a three-dimensional space that denies the possibility of a "stage" area; a perpendicular projection surface created from suspended fabric dissects the space and requires the performer and gallery visitors to navigate the space in order to perceive different elements of the work. Central to this version is the idea of an emergent performance event, that is, a performance that materializes without the rituals that mark the formal beginning and ending of a performance. The installation functions as before, but the opportunity exists for an instrumentalist to interact with the installation for a few moments, triggering audio and video events that may not have appeared otherwise or that may have been left to a random process.[6]

The first realization of this version took place over several hours, with visitors coming and going as they pleased and with performance events happening periodically over that time. The improvisational "performances" were not structured to create a musical whole in a conventional sense (though a certain amount of musical structuring by the instrumentalist and listener is probably inevitable); rather, the musical gestures were relatively short, discontinuous, and unevenly spaced over time. Gallery visitors were few enough that it was possible to engage with them in brief conversation before playing the instrument and between gestures, and to consciously disallow performance conventions to establish the event as a conventional performance. Still, this hybrid environment raised a number of ontological questions. In what ways do these musical interactions with the installation remain "performances"? Is a musician interacting with the installation remain a "performer" in the conventional sense, and does this interaction impose a different role on a casual gallery visitor witnessing this interaction? Can this mode of presentation of an audiovisual work transcend in some ways the limitations of conventional performance and installation?

photo credit: Sigi Torinus

As the creator and performer of the piece, I found the hybrid installation environment to be the most satisfying of the three versions; the unpredictability of certain aspects of the installation response made for an engaging aesthetic experience not dissimilar to improvising with other musicians, the video triggering engendered a sense of connection to the visual aspect of the installation, and the freedom to move around the space encouraged a more contemplative approach to the work. I was also pleased to hand over aspects of control of the listening experience to the gallery visitors; they could stay as long as they wished, they could be close to me or far away from me if they chose, and they could chat with each other without fear of interrupting a formal performance. Nonetheless, interacting with the installation while others in the space watched and listened seemed to ineluctably take on some aspects of a formal performance: I was privileged in that only I had access to the installation interface (in this case, the microphone), the sounds that I made for better or worse imposed themselves on the experience of others, and, perhaps in part due to the brevity of the emergent performance events, no gallery visitors left the space while I was playing. At the same time, no one applauded when I stopped playing, and gallery visitors for the most part continued to move around the space as they pleased.

The gallery visitors experienced some unease with the emergence of performance events; for some the proximity of the instrumentalist was disturbing, and responses varied from seeking a refuge on the periphery of the space to clowning with the installation. One visitor reported a discomfort with her perception that she was becoming an element of the work, in part because of the ongoing documentation taking place with multiple cameras. It was clear that the space that the audience normally inhabits in a performance had been transgressed by the instrumentalist, forcing a new and in some ways confusing role on the gallery visitors. All the same, those visitors that stayed for some length of time became accustomed to the occasional musical gestures, and within

[6] An interesting parallel exists in the performances of dada artists in the early part of the twentieth century. The dada artists often performed with little consideration of the audience, and were more preoccupied with their own experience within the art work and with the experience of other members of their group. This idea is explored in Annabelle Melzer's extensive research relating to dada performance practices. [6]

minutes had resumed the behaviours typical of casual visitors to an art gallery.

While I had spent much time anticipating the reaction of gallery visitors to the hybrid installation experience, I was surprised to find that the situation also posed some challenges for the musician that I had not anticipated. As the musician, I found it was difficult at first to identify an appropriate attitude to take in making the musical gestures. Should I present these gestures with intensity of a musical performer, or should the gestures be made more casually with the playfulness of an engaged participant in an interactive installation? Neither intensity nor playfulness seemed to reflect the nature of the work, so I adopted an attitude of detached experimentation and tried to approximate a random choice of gestures. This seemed to work well, though it became clear that future presentations of the piece with different musicians might require some discussion of this issue as part of an orientation to the work.

4. CONCLUSIONS

Ultimately, the hybrid mode of presentation seems to hold significant potential for an enriched experience of an interactive audiovisual work, both for the improvising musician and for the gallery visitors. From a visitor's perspective, the situation is in some ways akin to watching someone with considerable skill play a sport or a video game; there is an element of performance, but the "performer" is engaged with the activity in a way that is less directed at the observers and more with the game itself. This aspect will be developed more in future work.

Acknowledgments
This is an important message for Julie Wade was largely developed at the Noiseborder Multimedia Performance Lab at the University of Windsor, with support from the Social Sciences and Humanities Research Council of Canada. The performance version of the work was developed during a residency at the Residency Eina Danz in Norway. I gratefully acknowledge their support.

5. REFERENCES

[1] M. Carlson, Performance: A Critical Introduction, 2nd Ed., Routledge, 2003.

[2] A.R. Petersen, Installation Art: Between Image and Stage, Copenhagen: Museum Tusculanem Press, 2015.

[3] L. Ginters, "On Audiencing: The Work of the Spectator in Live Performance," in About Performance, 2010, pp. 7-14.

[4] G. Ouzounian, "Sound Art and Spatial Practices: Situating Sound Installation Art Since 1958," Ph.D. dissertation, Dept. Music, Univ. California, San Diego, 2008.

[5] D. Saltz, "The Art of Interaction: Interactivity, Performativity, and Computers," in The Journal of Aesthetics and Art Criticism, Vol. 55., No. 2, 1997, pp. 117-127.

[6] A. Melzer, Dada and Surrealist Performance, UMI Research Press, 1980.

A MODEL SELECTION TEST FOR FACTORS AFFECTING THE CHOICE OF EXPRESSIVE TIMING CLUSTERS FOR A PHRASE

Shengchen Li, Simon Dixon
Queen Mary University of London
shengchen.li@hotmail.com
s.e.dixon@qmul.ac.uk

Dawn A. A. Black
Radioscape
dawn.black@radioscape.co.uk

Mark D. Plumbley
University of Surrey
m.plumbley@surrey.ac.uk

ABSTRACT

We model expressive timing for a phrase in performed classical music as being dependent on two factors: the expressive timing in the previous phrase and the position of the phrase within the piece. We present a model selection test for evaluating candidate models that assert different dependencies for deciding the Cluster of Expressive Timing (CET) for a phrase. We use cross entropy and Kullback Leibler (KL) divergence to evaluate the resulting models: with these criteria we find that both the expressive timing in the previous phrase and the position of the phrase in the music score affect expressive timing in a phrase. The results show that the expressive timing in the previous phrase has a greater effect on timing choices than the position of the phrase, as the phrase position only impacts the choice of expressive timing in combination with the choice of expressive timing in the previous phrase.

1. INTRODUCTION

In classical music, performers vary the lengths of beats throughout a performance while keeping the overall beat rate. Such small variations of beat timing are known as expressive timing. Expressive timing contributes to the formation of expressiveness in classical music. Research into expressive timing shows that the expressive timing within a phrase is not randomly distributed but similar timing profiles are used across different phrases. There are various investigations about how such similar timing profiles can be found and how such common timing profiles are used by performers.

It is common in the literature to cluster the expressive timing in performed classical piano music into different types, with various temporal units used. For example, Repp [1] uses principal component analysis to analyse the commonalities and differences in performances of a Chopin Étude. Spiro et al. [2] use a self-organising map to cluster the expressive timing within a bar and investigate how the clusters of expressive timing are distributed. With model selection tests, Li et al. [3] demonstrate that clustering the expressive timing within a phrase is helpful for analysing expressive timing. Moreover, Li et al. [3] also introduce a method to cluster the expressive timing within a phrase by using a Gaussian mixture model. In this paper, we make use of model selection tests to show how the choice of Cluster of Expressive Timing (CET) is possibly affected.

There have been a few attempts to determine how expressive timing varies in a segment of performance. In [4] and [5], Widmer et al. discuss how expression in performed music is formed when the musical score is given. Their basic idea for expressiveness synthesis is to render each phrase using expressive gestures extracted from performances of similar phrases in a training database. In [4], the authors suggest that a dynamic Bayesian network may be used for expressiveness synthesis, in which case, the expressive timing in the previous parts of performance may affect the expressive timing in later parts. Similarly, in [6], Todd points out that parabolic curves can be used for fitting tempo variations across different levels of music structure. This suggests that tempo variations within a phrase can be affected by expressive timing in previous parts. Moreover, in the rule based system from KTH [7], the expressive timing is affected by both the music score and the sequence of expressive timing. As a summary of the works mentioned above, the music score and expressive timing in previous parts may affect the current choice of expressive timing.

In this paper, we examine two possible factors affecting the choice of CET for a phrase: the position of the phrase and the CET used in the previous phrase. In particular, we use model selection tests to demonstrate how the CET in a phrase is affected by both the CET used in the previous phrase and the position of the phrase in the musical score. We propose four Bayesian graphical models that assert different relationships between the CET used in a particular phrase, the CET used in the previous phrase and the position of the phrase. Then we design a model selection test to evaluate how well the candidate models predict the use of CETs. As the candidate models have different structures, we use cross entropy and Kullback Leibler (KL) divergence to evaluate the resulting models. Cross entropy and KL divergence are both derived from information theory and can evaluate models in different model spaces.

To obtain the CET distribution in this analysis, we follow the procedure developed in previous work [3], and use the same database: two Chopin Mazurkas (Op.24/2 and Op.30/2) and *Islamey* by Mily Balakirev [8]. In each candidate piece, the phrase lengths are identical throughout the piece. In addition, the beat timing for the two Mazurkas

is provided in the Mazurka database, which is used in various works [2, 9] by the CHARM group. The number of CETs varies from piece to piece according to our published methodology [3].

This paper is organised in the following way: we firstly introduce how the expressive timing patterns within a phrase are clustered. Then we observe how the CETs are distributed across different performers throughout a piece of music. Next we introduce the candidate models in this paper. Then we present the evaluation of the candidate models, followed by a discussion and conclusion.

2. CLUSTERING OF EXPRESSIVE TIMING

In this section, we describe how expressive timing is clustered in this work. For two Chopin Mazurkas, the tempo data is provided by the database. For *Islamey*, only beat timing is provided. We now introduce how we convert beat timing to tempo data for *Islamey* database. If we use t_i to represent the beat timing of the ith beat, for a piece of music that has n beats, the beat timing can be represented as $t_1, t_2, \ldots, t_n, t_{n+1}$ where t_{n+1} represents the ending time of the last beat in the piece. We use the reciprocal of the inter beat interval to represent tempo at the beat level (i.e. $\tau_i = \frac{1}{t_{i+1}-t_i}$). As mentioned above, the candidate pieces have constant phrase lengths throughout the piece, thus the expressive timing within the ith phrase that has w beats can be represented as $\mathbf{T_i} = (\tau_1, \tau_2, \ldots, \tau_w)$. By the expectation maximisation method, we can fit the distribution of expressive timing within a phrase to a Gaussian mixture model such that:

$$p(\mathbf{T_i}) = \sum_{a=1}^{A} \pi_a \mathcal{N}(\tau_i | \mu_a, \mathbf{\Sigma}_a^{full}), \quad (1)$$

where there are A clusters available, each with mean μ_a, covariance $\mathbf{\Sigma}_a^{full}$, and weight π_a for index a. If we use T_i^* to represent the CET that the expressive timing in phrase i belongs to, we have:

$$T_i^* = arg_a \max \pi_a \mathcal{N}(\tau_i | \mu_a, \mathbf{\Sigma}_a^{full}). \quad (2)$$

As discussed in previous work [3], the optimum number of CETs for a phrase differs from piece to piece. Using cross validation tests, the optimum number of CETs for the candidate pieces was found to be 2 clusters for *Islamey*, 8 clusters for Mazurka Op.24/2 and 4 clusters for Mazurka Op.30/2 [3].

Suppose that there are n phrases in a candidate piece of music and there are m performances in the database. If we use a vector to represent the clusters of expressive timing used for each phrase in performance j, we have $\mathbf{P_j^*} = (T_{1j}^*, T_{2j}^*, \ldots, T_{nj}^*)$. Thus we can use a matrix $\mathbf{P}^*$ whose row is $\mathbf{P_j^*}$ for performer j to represent the clusters of expressive timing used in each phrase for all performers. For easier observation, we convert matrix $\mathbf{P}^*$ to a diagram so that each element in $\mathbf{P}^*$ is represented by a colour block according to the cluster of expressive timing used. This type of diagram is called a Tempo Variegation Map (TVM) [10]. In Figure 1, we give a TVM for Mazurka Op.24/2 as an example. In this diagram, each row represents a performance of the Mazurka and each column represents a phrase. Each colour block represents a CET used in a phrase. The colours of blocks are selected according to the centroids of the CETs, with similar colours representing those clusters whose centroids are similar. By observing the distribution of the CETs, we propose the candidate models in this work.

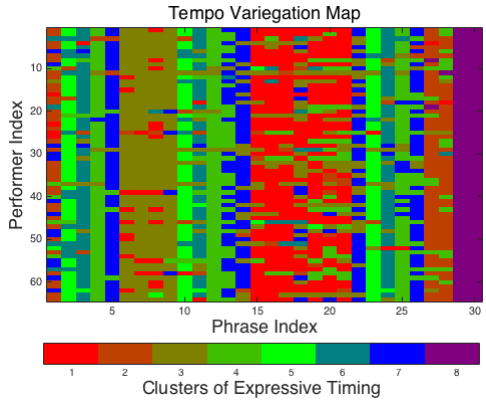


Figure 1: An example of a Tempo Variegation Map (TVM) for Mazurka Op.24/2.

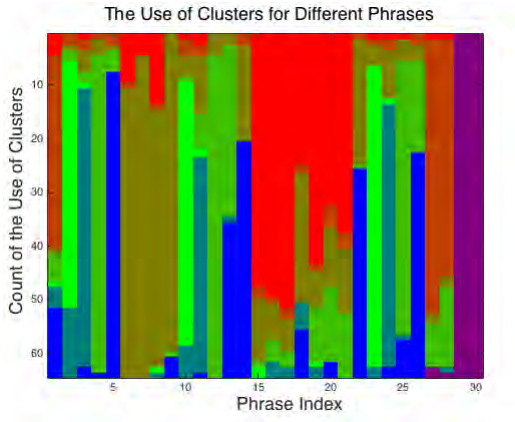


Figure 2: The distribution of CET used by all the performers for each phrase in Mazurka Op.24/2. The colours match those in Figure 1.

3. CANDIDATE MODELS

In this section, we introduce the four candidate models we propose according to our observations of the TVMs. To illustrate the observations for the candidate models, we use Figure 1 as an example. Then we introduce some regularities of the distribution of CETs and give the mathematical descriptions of the candidate models.

In Figure 1, we can see that for some phrases, the use of CET agrees across different performers. If we count the frequency of each CET for each phrase in a performance, we obtain Figure 2. In this figure, we see how the frequency of CETs differs from phrase to phrase, thus we propose the *positional model* (PM), which asserts that the position of the phrase in the music score affects the choice

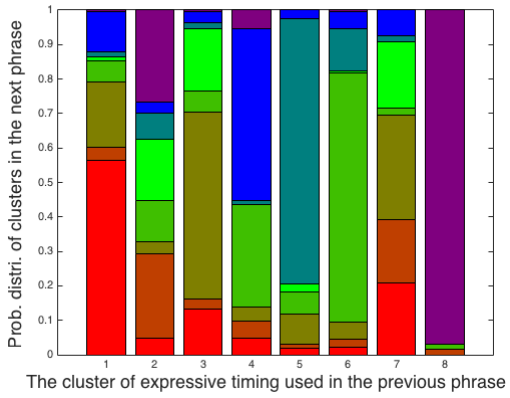


Figure 3: The relative frequency of CETs used by all the performers after a specific cluster is used in Mazurka Op.24/2. The colours match those in Figure 1.

of CET for a phrase.

Furthermore, if we observe Figure 1 again, we can see that some CETs are likely to be followed by a particular CET. For example, in Figure 1, cluster 5 is likely to be followed by cluster 6. If we visualise the relative frequency of each CET that appeared after another CET, we obtain Figure 3 that visualises the different distribution of CETs after a particular CET is used in the previous phrase. As a result, we propose the *sequential model* (SM), which asserts that the choice of CET for a phrase is affected by the CET used in the previous phrase.

Beside the positional model and the sequential model, we propose two other candidate models: a joint model and an independent model. The *joint model* (JM) asserts that the choice of CET for a phrase is affected by both the position of phrase and the CET used in the previous phrase. The *independent model* (IM) is a reference model which asserts that neither the position of phrase nor the CET in the previous phrase have any effect on the choice of CET for the next phrase.

In the candidate models, there are three variable parameters: the CET used in a particular phrase (T_i^*), the position of the phrase (β) and the CET used in the previous phrase (T_{i-1}^*). All candidate models are Bayesian graphical models that can be extended to a joint probability distribution of the parameters in the candidate models (namely $p(T_{i-1}^*, T_i^*, \beta)$). In the model selection test we use a cross-validation method to randomly select rows in $\mathbf{P}^*$ to form a testing dataset, with the remaining data in $\mathbf{P}^*$ forming the training dataset. We use five-fold cross validation to evaluate the candidate models. To remove the possible effects of the random train/test split, we repeat the five-fold cross validation tests several times.

Each training dataset is trained for finding $p(T_{i-1}^*, T_i^*, \beta)$ in the testing dataset. Then we evaluate how successfully $p(T_{i-1}^*, T_i^*, \beta)$ from the testing dataset is predicted according to the training dataset. The results derived from different formations of testing and training datasets are averaged. In some cases, certain combinations of $(T_{i-1}^*, T_i^*, \beta)$ may be absent in the training datasets but appear in the testing datasets. This will cause a problem of zero probability [11, Ch.17]. We use Bayesian estimation to learn the parameters in the candidate models to prevent the zero probability problem, which adds a small count to all probabilities [11, Ch.17]. For example, if there are x_1 samples such that $X = 1$ in a database that has x samples, the probability of $X = 1$ is defined by Bayesian estimation as:

$$p(X = 1) = \frac{x_1 + \frac{1}{x}}{x + 1}. \tag{3}$$

With the rule of multiplication for probability (if event A and event B are independent, $p(A, B) = p(A)p(B)$), Equations (4), (5), (6), and (7) define how $p(T_{i-1}^*, T_i^*, \beta)$ is calculated according to the Independent Model (IM), Positional Model (PM), Sequential Model (SM) and Joint Model (JM) respectively.

$$p_{\text{IM}}(T_{i-1}^*, T_i^*, \beta) = p(T_{i-1}^*) \times p(T_i^*) \times p(\beta) \tag{4}$$

$$p_{\text{PM}}(T_{i-1}^*, T_i^*, \beta) = p(T_{i-1}^*) \times p(T_i^*|\beta) \tag{5}$$

$$p_{\text{SM}}(T_{i-1}^*, T_i^*, \beta) = p(T_{i-1}^*|T_i^*) \times p(\beta) \tag{6}$$

$$p_{\text{JM}}(T_{i-1}^*, T_i^*, \beta) = \frac{\mathit{Count}(T_{i-1}^*, T_i^*, \beta) + \frac{1}{N}}{N + 1} \tag{7}$$

where N is the number of samples.

4. MODEL EVALUATION

The parameters in the candidate models are trained with the training datasets. Then we design a set of model selection tests to evaluate the candidate models to investigate how CETs are affected. In previous work [3] we demonstrated that model selection tests can be used for expressive timing analysis, using the Akaike Information Criterion (AIC) and the Bayesian Information Criterion (BIC). However, the AIC and BIC are designed to compare candidate models in the same model space [12, Ch.2-3], so in this paper, we use a more general method that compares the joint probability of $(T_{i-1}^*, T_i^*, \beta)$ in both training and testing datasets. The parameters used for model evaluation include cross entropy and KL divergence.

To distinguish the joint probability distribution in the training datasets and the testing datasets, we use $q(T_{i-1}^*, T_i^*, \beta)$ to represent the joint probability in the training dataset and $p(T_{i-1}^*, T_i^*, \beta)$ to represent the joint probability distribution in the testing dataset. For simplicity, we have $q_i \equiv q_{klm} \equiv q(T_{i-1}^* = k, T_i^* = l, \beta = m)$ and $p_i \equiv p_{klm} \equiv p(T_{i-1}^* = k, T_i^* = l, \beta = m)$.

The cross entropy between P and Q measures how many bits on average are required to code a symbol in P if we have a coding system whose probability distribution of symbols is given by Q. The cross entropy [13] for a distribution of n symbols is defined as:

$$H_{\text{Cross}}(P, Q) = -\sum_{i=1}^{n} p_i \log_2(q_i). \tag{8}$$

KL divergence, or relative entropy, is an indicator of how different a probability distribution Q is when compared to probability distribution P. The KL divergence is not a strict distance measurement, due to the fact that it is not symmetric ($KL_{Div}(P,Q) \not\equiv KL_{Div}(Q,P)$). The KL divergence is equivalent to the difference between cross entropy and the entropy of the testing dataset, as Equation (9) shows. In other words, KL divergence measures how efficient the coding system optimised for Q is for coding P.

$$\begin{aligned} KL_{Div}(P,Q) &= \sum_{i=1}^{n} p_i \log_2(\frac{q_i}{p_i}) \\ &= \sum_{i=1}^{n} \{p_i \log_2(q_i) - p_i \log_2(p_i)\} \\ &= H_{\text{Cross}}(P,Q) - H(P) \end{aligned} \tag{9}$$

5. RESULTS

In this section, we use two model selection criteria: cross entropy (defined in Equation (8)) and KL divergence (defined in Equation (9)). The model selection tests are applied with three candidate pieces: *Islamey* and two Chopin Mazurkas (Op.24/2 and Op.30/2). Following the method of Li et al. [3], the numbers of CETs used for analysis are 2, 8 and 4 for *Islamey*, Chopin Mazurka Op.24/2 and Op.30/2 respectively.

Criterion \ Model	IM	PM	SM	JM
Cross Entropy	7.25	7.71	7.12	**6.88**
KL Divergence	1.00	1.46	0.88	**0.63**

(a) *Islamey*

Criterion \ Model	IM	PM	SM	JM
Cross Entropy	10.63	13.31	9.69	**7.74**
KL Divergence	4.22	6.90	3.24	**1.36**

(b) Chopin Mazurka, Op.24/2

Criterion \ Model	IM	PM	SM	JM
Cross Entropy	5.80	6.60	5.60	**4.92**
KL Divergence	1.69	2.49	1.49	**0.81**

(c) Chopin Mazurka, Op.30/2

Table 1: Model evaluation of the candidate models that assert different dependencies on the CET used in a phrase. Both model selection criteria use a smaller value to indicate better model performance. The IM, PM, SM and JM are defined in Section 3. The bold value indicates the best performance of the candidate models.

For all candidate pieces, we use five-fold cross-validation to test the candidate models. For a single experiment, we select the data from 20% of performances in our database randomly to form the testing dataset and the remaining 80% of performances forms the training dataset. The experiment is repeated 100 times to mitigate the possible effects of randomness in forming testing and training datasets. The results of the model selection criteria for evaluating how well the testing datasets are predicted are then averaged to obtain the final results.

Criterion \ Model	IM	PM	SM	JM
Cross Entropy	7.47	9.07	**7.27**	9.05
KL Divergence	0.92	2.56	**0.77**	2.52

(a) *Islamey*

Criterion \ Model	IM	PM	SM	JM
Cross Entropy	11.07	13.72	**11.01**	11.06
KL Divergence	4.26	6.90	**4.15**	4.39

(b) Chopin Mazurka, Op.24/2

Criterion \ Model	IM	PM	SM	JM
Cross Entropy	**6.11**	6.60	6.29	6.56
KL Divergence	**1.80**	2.22	1.96	2.13

(c) Chopin Mazurka, Op.30/2

Table 2: Average model selection criteria for a training dataset of only one performance. The IM, PM, SM and JM are defined in Section 3. The bold value indicates the best performance of the candidate models.

In Table 1, we present how well the candidate models predict the testing dataset on average. According to the cross entropy and the KL divergence, the joint model is the best model among the candidate models. The sequential model is the second best model. The positional model is even worse than the independent model.

Data-size robustness means how much the model performance drops when a very limited amount of training data is available. The data-size robustness is a property of the candidate model. In this paper we compare the results obtained using 80% of performances for training (Table 1) with those obtained using only one performance for training (Table 2).

From Table 2, we notice that the data-size robustness of candidate models varies for different candidate pieces. For *Islamey* and Mazurka Op.24/2, the best model in terms of data-size robustness is the sequential model. For Mazurka Op.30/2, which has a training dataset of only 7 phrases, compared with 39 and 29 phrases for *Islamey* and Mazurka Op.24/2 respectively, no model performs better than the baseline independent model. Between the sequential model and the joint model, which have the lowest cross entropy and KL divergence in the model selection tests, the sequential model is more data-size robust.

In summary, the joint model is the best model when a reasonable amount of data is available for training. When the amount of training data is limited, the sequential model is preferred as it is more data-size robust with acceptable results. Based on the model selection tests, we demonstrate that both the position of phrase in music score and the CET used in the previous phrase affect the decision of CET for a phrase. The CET in the previous phrase has a greater effect than the position of the phrase, which only affects

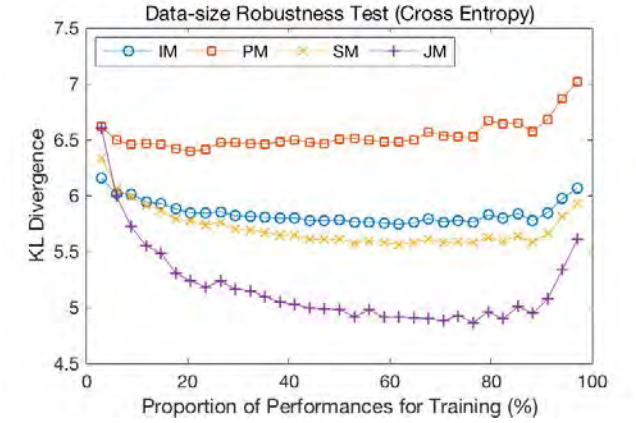
the choice of CET for a phrase jointly with the CET used in the previous phrase.

6. DISCUSSION

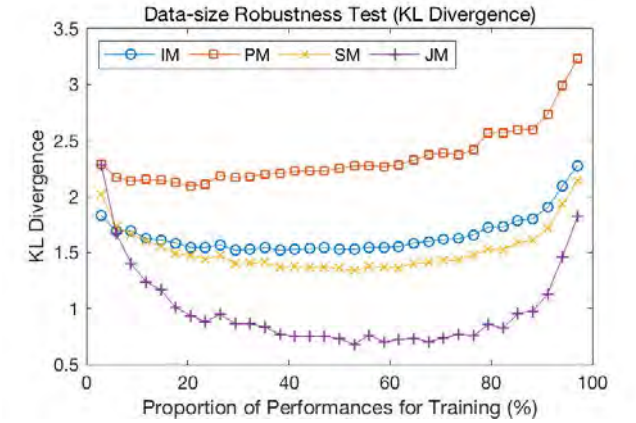
6.1 Comparison of model selection criteria

The results for training with 80% of performances (Table 1) are in some cases worse than those obtained by training on only one performance (Table 2). This fact suggests that using only 1 performance for training results in a better model than using 80% of performances for training in these cases, which conflicts with the intuition that having more data for training usually results in a better model. As a result, we investigate the data-size robustness in more detail.

In Figure 4, we show how cross entropy and KL divergence of candidate models vary with the proportion of performances used for training. Because of the page limitation, we show results only for Chopin Mazurka Op.30/2 as an example in Figure 4. The other pieces give similar results.



(a) Cross Entropy



(b) KL Divergence

Figure 4: Model evaluation as a function of the percentage of performances used for training. The test piece is Chopin Mazurka Op.30/2. A larger number indicates a poorer model.

Observing Figure 4b, the KL divergence start to increase when more than about 70% of performances are used for training. This leads to a higher KL divergence for training sets of 80% of performances than for only one performance for training (compare Tables 1 and 2). In addition, the distribution of testing data can be heavily biased such that even a good model may have a high KL divergence. As a result, model selection tests with KL divergence may be affected by the bias of the testing dataset, whereas the cross entropy test is less affected by bias in the testing dataset.

Usually, if there are more data available for training, we expect the resulting model to be better. However, in Figure 4, the curves of KL divergence and cross entropy are not as monotonic as we expect. If we calculate the first order difference of the cross entropy and KL divergence changes, we find KL divergence has a lower zero-crossing rate than the cross entropy ($p = 0.0161$). In other words, the curves of KL divergence are "smoother" (mathematically speaking, more monotonic) than the curves of cross entropy. The non-monotonic changes in both KL divergence curves and cross entropy curves may be caused by the randomness of the dataset as we have only tried 100 out of the possible 10^{15} formations of the training dataset in this experiment.

Comparing the two model selection criteria used, KL divergence appears to be less sensitive to the randomness of testing data but more sensitive to bias in the testing data, whereas cross entropy is sensitive to the randomness of testing data but is less sensitive to bias. As both model selection criteria agree on the ranking of results, the conclusions in this paper should be robust against the effects of both the randomness and bias of the testing datasets.

6.2 Model complexity

Despite our results showing the rank of candidate model as joint model, sequential model, independent model and positional model, the model complexity of the candidate models do not follow the same order. As there are 2, 8 and 4 CETs chosen in the analysis for *Islamey*, Chopin Mazurka Op.24/2 and Op.30/2 respectively, the number of parameters to be trained in candidate models are listed in Table 3. The number of parameters are decided by the number of phrases in the pieces as well, which is also listed in Table 3. We find that despite the high complexity, the joint model outperforms the other models. The sequential model is more data-size robust due to lower complexity. The results of the model selection tests presented in this paper support the claim that the choice of CET in a phrase is primarily affected by the CET used in the previous phrase. Moreover, the position of phrase in the score only affects the CET in a phrase jointly with the CET used in the previous phrase. Further investigations are required to understand why the positional model alone performs worse than the baseline independent model.

6.3 Future work

In this paper, we present a model selection test that investigates how the CET used in a phrase is affected by the CET used in the previous phrase and the position of a phrase. However, with the same methodology, we can demonstrate how other musical features affect the choice of CETs. The

Pieces	# of phrase	IM	PM	SM	JM
Islamey	40	2	80	4	160
Op.24/2	30	8	240	64	1920
Op.30/2	8	4	32	16	128

Table 3: Number of parameters learnt in candidate models. Abbreviations of models are defined in Section 3.

position of phrase is a simplistic concept which gives little insight into the reasons for expressive choices. There are multiple features in a phrase related to the melody, harmony and rhythm which are likely to influence the performer's choices. For both expressiveness synthesis and musicology research, it would be interesting to investigate further which factors derived from the musical score affect the choice of CET.

Likewise, the exclusive use of the previous CET is a simplification of possible longer term temporal dependencies which may exist between expressive choices, including even non-causal (i.e. planned) relationships with future choices. While the local context is likely to have the largest influence on immediate choices, it would be naive to assume that expert musician's timing choices can be modelled by a simple first-order process.

This research is based on a clustering method proposed in previous work [3]. This method has strong restrictions: the phrase length throughout the candidate piece must be constant, and the number of CETs used varies according to the candidate piece. These restrictions prevent the immediate application of the proposed method to a wider range of performances and a larger dataset using the current clustering algorithm. Further research is required to extend the methods to more general scenarios. Finally, applying the experiments in this paper with generalised model selection methods, such as AIC and BIC, may demonstrate how the model complexity affects the model selection process.

7. CONCLUSIONS

In this paper, we presented a model selection test that investigates how the Cluster of Expressive Timing (CET) is chosen according to the position of the phrase and the CET used in the previous phrase.

We proposed four candidate models that assert different dependencies of the CET used for a particular phrase. We evaluated the four candidate models with KL divergence and cross entropy. The results of the model selection showed that the joint model is the most reasonable model for selecting the cluster of expressive timing for a phrase. However, if there are only very limited data available, the sequential model should be used, owing to its lower complexity.

Hence we have shown that both the CET used in the previous phrase and the position of the phrase affect the selection of CET for a phrase. The sequence of clusters has a greater effect than the position of phrases for selecting the cluster of expressive timing for a phrase. The position of the phrase, on the other hand, only affects the choice in combination with the CET sequences.

8. REFERENCES

[1] B. H. Repp, "A microcosm of musical expression. I. Quantitave analysis of pianists' timing in the initial measures of Chopin's Etude in E major," *The Journal of Acoustical Society of America*, vol. 104, pp. 1085 – 1100, 1998.

[2] N. Spiro, N. Gold, and J. Rink, "The form of performance: Analyzing pattern distribution in select recordings of Chopin's Mazurka op. 24 no. 2," *Musicae Scientiae*, vol. 14, no. 2, pp. 23–55, 2010.

[3] S. Li, D. A. A. Black, and M. D. Plumbley, "Model analysis for intra-phrase tempo variations in classical piano performances," in *Proceedings of Computer Music Multidisciplinary Research (CMMR'15)*, 2015.

[4] G. Widmer, S. Flossmann, and M. Grachten, "YQX plays Chopin," *AI Magazine*, vol. 31, no. 3, pp. 23–34, 2010.

[5] A. Tobudic and G. Widmer, "Relational IBL in music with a new structural similarity measure," in *Proceedings of the 13th International Conference on Inductive Logic Programming (ILP'03)*. Springer, 2003, pp. 365–382.

[6] N. P. M. Todd, "The dynamics of dynamics: A model of musical expression," *Journal of Acoustical Society of America*, vol. 91, pp. 3540–3550, 1992.

[7] A. Friberg, R. Bresin, and J. Sundberg, "Overview of the KTH rule system for musical performance," *Advances in Cognitive Psychology*, vol. 2, pp. 145–161, 2006.

[8] M. Balakirev, *Islamey, Op. 18*. Hamburg: D. Rahter, 1902. [Online]. Available: http://imslp.org/wiki/Islamey,_Op.18_(Balakirev,_Mily)

[9] C. Sapp, "Hybrid numeric/rank similarity metrics for musical performance analysis," in *Proceedings of the International Conference on Music Information Retrieval (ISMIR)*, 2008, pp. 501–506.

[10] S. Li, D. A. A. Black, E. Chew, and M. D. Plumbley, "Evidence that phrase-level tempo variation may be represented using a limited dictionary," in *Proceedings of International Conference on Music Perception and Cognition (ICMPC'14)*, 2014.

[11] D. Koller and N. Friedman, *Probabilistic Graphical Models: Principles and Techniques*. The MIT Press, 2009.

[12] G. Claeskens and N. L. Hjort, *Model selection and Model Averaging*. Cambridge University Press, 2008.

[13] J. E. Shore and R. W. Johnson, "Axiomatic derivation of the principle of maximum entropy and the principle of minimum cross-entropy," *IEEE Transactions on Information Theory*, vol. 26, pp. 26 – 37, 1980.

The Sound Bubble: An Aesthetic Additive Design Approach to Actively Enhance Acoustic Office Environments

Martin Ljungdahl Eriksson
Media and Design
University West
46132 Trollhättan, Sweden
martin.ljungdahl-eriksson@hv.se

Lena Pareto
Media and Design
University West
46132 Trollhättan, Sweden
lena.pareto@hv.se

Ricardo Atienza
Konstfack, University College of Arts, Crafts and Design
126 27 Stockholm, Sweden
ricardo.atienza@konstfack.se

ABSTRACT

Moving towards more open and collaborative workplaces has been an emerging trend in the last decades. This change has led to workers sharing a common open space, with seating's based on current activity, so called activity-based offices. Consequently, it becomes difficult to design sonic environments that cater to different needs in the same space. In this study we explored the possibility of adding site-specific but location-adaptive sound environments to enhance the experience of an activity-based office workplace. For this purpose, we developed the concept of the "sound bubble," a micro-space in which the user is embedded by a semi-transparent sound environment. The purpose of the bubble is to help the user ignore irrelevant and disturbing noise while working in an open landscape. The sound bubble supports the user to stay in "everyday listening" mode, i.e., not focusing on anything particular in the surrounding environment while being able to keep a link with it. The sound bubble was evaluated by a total of 43 test subjects participating in an experience-based test, conducting their usual work tasks in an office landscape. Our results show that the sound bubble can enhance auditory work conditions for individual work requiring concentration.

Author Keywords: acoustic design; sound design; sonic interactive design; sonic micro-milieu; sound bubble; site-specific designed ambience.

1. INTRODUCTION

In recent years, the research and practice field of sound design to improve every-day environments such as work places has experienced a substantial evolution. In an area traditionally dominated by physical acoustics, a wide range of new concepts and methods have emerged allowing site-specific solutions beyond the limited noise reduction approaches, e.g., insulation, absorption, noise cancelling, and energy masking. Also in the fields of sound studies and ambiences research, new tools and methods have been provided and explored for dealing with everyday complex public or private shared environments. This approach, like ours, takes on a constructive, creative approach and develops methods to manage sounds as mediators of qualitative information, as opposed to the traditional, defensive approach to only protect people from sounds. See for example the work by Laboratory CRESSON since the 90's and its application into different case studies [2] [13]. Getting a space as silent as possible is far from enough: every sonic diagnose and treatment needs to take into account the complexity of a whole physical, social and sensorial environment. Such holistic approach methodologies are based on fields such as psychoacoustics [20] and ambience theory [12]. For example, in psychoacoustics the concept of information or attention masking was introduced, which is an action not physically masking the environing sounds but redirecting or abstracting the user's attention from the environing context. In ambience theory, new methods such as the interdisciplinary sonic effects have emerged [2].

Our approach to deal with these complex environments is a constructive method where digitally constructed, subtle sound textures are added to the local ambience. Such textures are meant to consciously or unconsciously transform the perception and experience of the current sound environment. A similar approach was used to enhance the sound environment in trains [3], [5]. Here, the method is further developed, the context is indoor office landscapes, and the sound bubble solution is localized to individual usage as opposed to sound environments for whole train coaches.

In recent years, activity-based and flex-offices have been gaining in popularity. What characterizes these types of offices is that there is not a fixed workplace for each individual as in traditional offices, rather there are workplaces accustomed to different purposes common to all employees to choose from based on current activity [8]. Beyond an economic motivation, such organization acknowledges that many types of jobs require flexible spaces due to the variation of tasks required by office workers. To minimize noise interference is a complex challenge for acoustic design in such offices. The main source of noise interference is considered to be co-workers talking [16]. However, talk is not only a problem in open office landscapes; it can also lead to improved knowledge sharing and ease of communication between employees.

Research has proved that sound perception is emotionally conditioned: a general positive attitude towards the work environment results in greater tolerance for the acoustic environment [23]; sounds derived from

things we like are considered less disturbing [23]; sounds that we understand the meaning of and which we find useful disturb us less [17]. In addition, it has also been proved that constant noise disturbs us less than occasional, sudden noises [17] [19]. And of course the type of work to be performed will also affect our sensitivity.

The focus of previous research has primarily been on noise reduction and understanding the effect of such noise on working conditions while very little has been done in terms of how to actively improve such complex environments. According to [10] there is a need for more knowledge about how to integrate creative practices with contextual influences (users, environments, or activities) that are seen as key elements of situated design practices.

In this study we explore the possibility of improving the sound environment through inserting context-dependent, adaptive sonic textures in an activity-based office environment. We argue that an active (as opposed to passive traditional acoustic methods such as noise reduction, absorption and insulation) and a constructive design approach is required to significantly improve the sound environments in complex contexts such as open office landscapes.

2. APPROACH

Our aim is to improve the quality of office workspaces and our means is to design site-specific and location-adaptive sound environments. We have promising results from testing the concept in a laboratory setting where we found that the designed sound environment immersed the listener and generated a sensation of an encapsulating sound bubble [9]. In this study we take the research and design process one step further and evaluate a location-adaptive prototype in a real office context.

2.1 Research approach

Our overall approach is grounded in design-based research [24], which is a systematic but flexible methodology aimed to improve practices through iterative design interventions. Our approach is further grounded in acoustic design [14], which focuses on the treatment of sound environments in relation to their architectural design and acoustics. This implies treating sound as a positive design element to create environments that interact with all senses. Acoustic design suggests that the function of sound is to support the ongoing activity, i.e., it must be contextualized. Therefore, the field of contextual design [6] will also be relevant in our research process, a design methodology where user and context of use are essential components. Last but not least, and at the core of our methods and practices, sonic interaction design (SID), which is an emerging field that interweaves auditory display, interaction design, ubiquitous computing and interactive arts [18]. SID research is dependent on knowledge of everyday sound perception, ecological acoustics, and sound and music computing [18]. SID also aims to identify unconventional ways to use sound in the interaction between users and artefacts, services or environments [15].

3. RESEARCH QUESTIONS

We pose the following research questions; all devoted to the aim of exploring if and by which sounds a working environment could be improved by sound design:

1. *How do users perceive the acoustic environment with the sound bubble compared to without it?*
2. *How do users perceive and describe the sound bubble and their experiences with the different sounds in the bubble?*
3. *Which sound environments are preferred and used?*

4. THE SOUND BUBBLE CONCEPT

When using the concept of "bubble", it's important to clarify that we do not intend to isolate the worker by physically masking the environing sounds; far from that, our aim is to provide a porous semi transparent sound bubble able to help people focus on their work while maintaining a link with their working context and colleagues.

As a first theoretical frame, we initiated our research process and design concept in Pierre Schaeffer's aesthetics in Musique Concrète theory [7], in which sound perception is categorized into four different "Listening Modes": hearing, listening, comprehending and understanding. Hearing is the most elementary perceptual level where we passively take in sounds that we do not try to listen to or understand. Listening involves the collection of information, where we direct our aural attention to someone or something in order to identify the event. Comprehending however involves processing and selection of sounds, to choose what interests us, to qualify and react to the inherent properties of the sound. Understanding involves semantics where the sound is interpreted as a sign or code that represents something meaningful to us. Gaver [11] describes the concept of everyday listening where each person hears the environing sounds as surrounding events in a larger context and not as sounds to pay attention to per se. Another similar idea is suggested by [1] when describing our common sonic attention mode as "floating listening", i.e., perceiving without focusing our attention, keeping some kind of potential listening and finally reacting when unexpected, uncommon events emerge. In everyday listening mode or floating attention, we are aware on a more subconscious level of the origin of a sound. Where sounds get a function and can be related to e.g. an activity, everyday listening is then more of a response to hearing. We often unconsciously shift between these listening modes.

4.1 Design concept

Our design concept is intended to help users of the prototype to get into hearing mode or everyday listening mode in order to help them focus better on their job tasks. A way to attain that is to camouflage undesired talk by creating what [1] describes as a "sonic micro-milieu" that takes precedence over a distant or secondary perceptive field. The resulting dominant effect is that of perceptually

placing the inserted environment in front of the background sound.

For obtaining a non disrupting micro-milieu, a subtle modification on the existing environments was required. In order to achieve that, different techniques were tested and evaluated such as space and time manipulation of the existing sounds, as well as insertion of other sonic contexts and materials; the goal being to generate a micro-milieu not invoking any specific sonic attention, but providing a somehow "natural" environment for such context and tasks to accomplish. The sound components should be abstract enough –e.g., non musical– so that most people do not recognize them as being familiar or analyse them in terms of musical taste for example. [22] investigated in a study the ease of learning different sound types and found that abstract sounds were learned and retained with far greater difficulty than both speech and representational sounds. A contradictory condition defines thus the nature of the sounds we are to design: on one hand they should be "obvious" enough not to attract prolonged attention; on the other hand they should not be recognizable as that may also lead to focused attention. An interesting paradox we are trying to solve.

The sounds are therefore designed to blend into the environment in order to be perceived as a continuous stream of similar sounds. The aim is to create a sonic micro-milieu triggering the hearing mode and ultimately everyday listening only. The design concept aims to facilitate interaction with the local sound environment by adding a semi-transparent sound environment. In the sound bubble the user is embedded by sounds that should not require sonic attention. This makes it possible for the user to select what to focus on, the sound in the prototype or the surrounding sound environment.

4.2 Individual and collaborative models

Two basic sound models have been developed and explored in this research, one for individual concentrated work and one for creative, collaborative work:

i) Based on acoustic theory and related work, our first design concept for concentrated individual work aimed at creating sound atmospheres that could be perceived as spatially confining, soothing and enhancing an inward attention.

ii) The second sound design concept, for creative collaboration, aimed at creating sound environments that could induce the experience of space and motion, sound textures that arise from random locations in the environing space bringing unexpected elements to metaphorically simulate –and stimulate– the idea of opening up the senses, being open to the unexpected, and thereby stimulate creativity.

5. THE SOUND BUBBLE SOUNDS

The sound bubble sounds were designed, tested and refined iteratively in two phases: First two sound concepts where iteratively designed, tested and refined in a laboratory setting. Then, these two concepts were implemented into 2 variants of sound environments from each concept. Also, a "neutral" sound was developed to be used as comparison in the study. The five sounds were then evaluated in an in-situ study with office workers.

5.1 Two Sound Concepts developed in laboratory

Our original idea was to generate sound sequences consistent with the physical environment, using for example sounds already present in the offices or subtly inserting filtered noise molded to the particular needs of the space and activities in question. However, the sounds needed to be modified in order to answer to the problems detected in these spaces: disturbing speech from other colleagues, background sounds from cooling systems, etc. For that, original sounds would be modified, e.g. by changing the frequency range, applying different perceptual effects (like inserting delay lines), creating certain rhythms or masking the speech by adding sounds able to absorb or flatten major variations in the original sound image. Three sounds were developed based on the two concepts and tested in the first laboratory phase. These sounds were refined into 5 final sounds evaluated in the main phase of the study.

5.2 Five Sounds Environments tested in-situ

The two design concepts were orchestrated into five sound environments developed for the in-situ test. Their purposes were twofold: 1) to have attention masking effects on surrounding talk and background noise as opposed to traditional energy masking, e.g., water sounds of a fountain for treating the sounds of traffic, and 2) to support intended activities in the office landscape.

To evaluate if there were any preferences of sound environments related to the intended activities in the office landscape we designed two sound environments for supporting focused work (A and B), and two for facilitating creative work (C and D). The fifth sound environment (E) was a recording of the background noise of an empty office. Sound E served as a static ambient background sound related to the room rather than to the activity. It was the closest we could get to a "placebo" in medical experimental settings, since no sound at all was too different to serve as a neutral, control condition.

The sound environments were based on results from previous research conducted by one of the authors concerning sound design of high-speed trains [3][5]. Those studies focused on different situations but deployed similar methods and aims as they were also exploring potential improvement of complex environments through the insertion of designed, site-specific sound environments. In the train context, sounds based on cycles and repetition proved to be the most pleasant ones compared to designs based on the notions of variation or improvisation. In that case, the idea of cycle was connected with rhythms of the human body (breathing) as well as sea rhythms.

Our expectation is that this type of sound environments would only be perceived in *hearing mode*, which means they needed to establish some kind of new background atmosphere not attracting too much attention. This could be achieved by composing as legato as possible, in the terms of Schlittmeier and Hellbrück [21]; their work

showed that legato –i.e., a sonic continuous stream characterized by few dynamic and spectral variations–, affected cognitive ability less than music with distinct temporal and spectral variations.

Grounded in our theoretical frame and derived from findings and insights from the train studies and the foregoing laboratory test; five site-specific and location-adapted sound environments were developed. The overall idea was as follows: The first 2, sounds, A and B, was intended to support focused work through the exploration of static sequences in terms of emerging sound events, dynamic progressions and timbre variations. Conversely, sounds C and D intended to promote creative work through more dynamic (even if still humble) characteristics; in order to reach that, these sequences were given a subtle discursive line, slowly evolving in register and timbre.

The sound bubble concept involves connecting the sounds with the particular location and making the sound bubble adaptive to the surrounding environment. This was achieved in this first prototype system by modifying recordings of the office ambiance and having changes in the amplitude of the surrounding environment trigger alterations in playback speed and panning of all the sounds, except in the placebo sound E that was static. The detailed design of each sound environment is described below.

Sound A: the purpose of sound A is to simulate the sound of sea waves and wind. It was generated by dynamically filtering pink noise. The filter was used to restrict the frequency range to attenuate certain frequencies of sound and alter the sound image. The filter allowed a softer and more appealing pink noise. To achieve a corporeal rhythm the pink noise was faded in and out following a breathing tempo, following similar ideas as in the train case. Finally, the modulation was panned, which meant that the sound slowly moved between the speakers to give the impression of a slow wave movement. Soft pink and white noise have traditionally been considered as a way to mask unwanted background noise; with this example, we wanted thus to test the validity of such protocol.

Sound B is based on the same concept and material as sound A, but here we aimed for a tone-based example instead of the noise character of sound A. For that, a resonator and a delay have been added. The resonator was used to reinforce a number of frequencies, affecting the timbre and at the same time changing the key signature in order to colour the pink noise and transform its perception into a chord-based resonance. The sound was split into two layers, the second layer pitched an octave above the first and delayed to maintain the sensation of a wave motion.

Sound C is derived from the other concept, that targeting collaborative, creative work. Sound C is based on an iterative 5th interval, in search of ever-changing tonal relations. Different chains of ascending 5th's, starting in non-related tones (not belonging to the same traditional scale in hz), generate a cyclical structure without clear reference points for the listener. The aim with a non-referential structure is bring about the sensation of an open, unpredictable space, and thereby induce a suspended floating listening mode.

Sound D is derived from the same concept as sound C. Sound D is based on a drone-based, static and coloured sound texture, which spectral envelope slowly evolves within a cyclical structure. It is created by applying different sound effects –equalizer, reverb, delay and resonator– on the recording of a walk on a pebble beach, obtaining thus a static sound texture where the corporeal rhythm of walking is still present as a background print.

Sound E is a recording of an empty office; our "placebo". This recording, belonging to an office space, was equalized and amplified for it to be noticeable as a distinct sound material within the local surrounding environment. In this case we insert a space within a space, both of similar character, exploring the masking potential of such confrontation.

The five sound environments had a length of 15 minutes each and seamlessly looped when played.

6. THE SOUND BUBBLE PROTOTYPE

In order to evaluate the sound environments, one prototype for the individual and one for the collaborative setting were developed. The individual prototype (Fig.1) consisted of an office chair on which speakers were attached, about 5-8 cm from each ear depending on how the user moved his/her head, at the headrest on two rods directed forward. A user could to some extent alter the preferred listening position because the speakers could be moved horizontally. Placing the speakers that close to the head resulted in an effect best described as the sensation of an encapsulating sound bubble.

Figure 1. The prototype: detailed view on the speakers.

A laptop was connected to the speakers. The laptop controlled audio playback and logged which sounds were played and for how long.

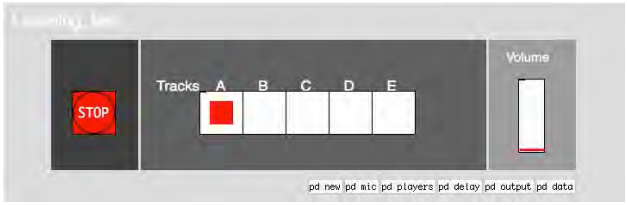


Figure 2. Interface.

The interface (Fig. 2) designed for the test was developed in the graphic programming environment PureData. The users were confronted to a very simple

window reduced to five boxes labelled A-E where it was possible to select the different sound environments (random distribution different for every user) and a slider to adjust amplitude level. Since the purpose was to investigate how added sound in an office landscape is perceived we chose not to enable the users to mute the sounds completely.

The prototype setup also contained a microphone that registered amplitude changes, which affected how sounds were processed in the laptop. We assumed that an increased amplitude level registered by the microphone would originate from activities, e.g. people moving in the immediate environment or conversations. Amplitude changes in the immediate environment affected the tempo and panning of the sound environments in the speakers. When the measured amplitude increased that led to a narrower stereo sonic space, and the sound bubble was then perceived as more compact and encapsulating. In parallel, an increase of the amplitude also generated a temporal acceleration in terms of playback speed.

A normal sound pressure level in offices is approximately 50 dB [20]. The Swedish Work Environment Authority, which is an administrative authority for occupational health and labour issues, has published a white paper about noise, AFS 2005: 16. According to that document a workplace with a sound pressure level of 50 dB (A) has satisfactory speech intelligibility. Therefore, we decided to set the limit for when the interface should react and alter tempo and panning at 50 dB (A).

The prototype for collaborative setting consisted of four speakers, one in each corner of a meeting room. A laptop with an external soundcard managed the four speakers. The interface was identical for both prototypes.

7. THE SOUND BUBBLE EVALUATION

The sound bubble was evaluated first in a laboratory setting with experts, then in-situ in an office landscape with office employees. The study design was piloted prior to the full-scale office test.

7.1 Laboratory test

A first trial run was performed in a lab environment to investigate the possibility to enhance the room ambience by actively adding sound, thereby creating a barely noticeable sound environment influencing the user. The trial run was conducted through three iterative phases:

- An *exploration phase* where the conceptual ideas and the first examples of the sound designs were developed.
- A first *office simulation test phase*, in which an in vitro test scenario was developed and presented with the visual support of video recordings from working spaces. Immersed in this virtual environment, two sound experts helped to evaluate our first sonic sketches.
- A second office *simulation experiment*, in which the sounds were slightly improved and incorporated different video stimuli; the experiment was conducted with the help of 4 work-environment experts and 3 open-office employees.

7.2 In-situ test location

The chosen location for the in-situ test was a table with seating for 10 people in an activity-based office at the IT department in a large manufacturing industry in West Sweden. Table occupants worked both individually and in groups. The table was characterized by spontaneity as several group meetings could take place simultaneously and sometimes meetings would end up in new meetings among team members. The table was located in front of three conference rooms that generated a stream of passing people. The immediate environment thus became a natural gathering place for spontaneous meetings and conversations. The location of the collaborative prototype was inside a frequently used meeting room with one large table for 10-15 people behind the abovementioned table.

7.3 In-situ Pilot study

Prior to the main field test, we conducted a pilot test during 8 days, to determine if the study design was suitable. 16 test persons performed self-selected work tasks while using the individual prototype. Which sonic environments the participants played, how long they were played as well as what amplitude level the user chose was logged. After the test, the participants answered a questionnaire, with background questions about gender, age and perceived hearing sensitivity, work-related questions and overall questions about the sounds and their experience of using the prototype.

7.4 In-situ Study

Forty-three test subjects were recruited to the experiment, which had undergone a relevant change: random assignment of sounds to interface buttons. Participants were asked to answer a questionnaire, which was slightly improved relative the pilot test. All participants were observed and they also had the opportunity to read through the observation protocol to comment and clarify any misunderstandings. The observer took notes on visual and auditory events and how the test subjects seemed to react to the events. Two test participants were recruited to use the prototype for an entire working day, and these test sessions were followed up with supplementary semi structured interviews. Questions included if they perceived the sounds differently during the day and why they choose the sounds.

The recruitment of test subjects for the collaborative setting was more difficult, since all meeting delegates had to accept having the sound environment running during the meeting. Also, observation in the meeting room was not allowed due to meeting confidentialities.

8. RESULTS

8.1 In-situ study

Of the 43 participants in the individual experiment, 12 were female and 31 were male, which reflects the employee ratio in the office. The participants ranged in age from 24-58 with a mean of 38,7. One person reported

impaired hearing. The collaborative experiment failed in getting enough participants to complete the test due to the abovementioned difficulties, and therefore are the results below derived from the individual experiment only. The research questions are answered as follows:

8.1.1 How do users perceive the acoustic environment with the sound bubble compared to without it?

In a direct question enquiry with the categories (better, same or worse), participants were asked how the sound bubble changed their sound environment. 74.4% answered better, 16.3% answered the same and 9.3% responded worse.

8.1.2 How do users perceive and describe the sound bubble and their experiences with the different sounds in the bubble?

To evaluate how the users perceived our designed sound environments we applied Axelsson's model from 2010 [29]. The model is based on several previous works since the 80's and provides pertinent criteria for the assessment of an environment. The participants were asked to evaluate the added sound environment they preferred in terms of 4 criteria-pairs, presented as 8 separated criteria: pleasant-unpleasant, eventful-uneventful, peaceful-chaotic, exciting-monotonous. The results below show the qualities characterizing the sound of each participant's preferred sound. The analysis shows which characterizing criteria the participants preferred.

Two criteria present a well-defined dominance, pleasantness and peacefulness, where participants clearly state a preference for their positive dimension *pleasant* and *peaceful* (see Figure 3, right). A participant described the experience in this way: "*The sound I chose was pleasant and I could ignore the [sonic] details, it did not change too much. The sound was very constant. […], it was in the background all the time. It was easier not to focus on it. The wind sound (A) was a bit too significant, I was thinking too much about the sound and too little on the job.*" The qualities pleasant and peaceful seem to be globally reinforced for the preferred sounds and in particular for sound 5 (in black). One participant described her experience as: "*A sonic wallpaper that gave a faint pleasant sound*", "*Silence in a wave noise*", and "*The sound felt both calm and gave focus*." A high correlation can be observed between pleasantness and peacefulness (see figure 3, rightmost graph).

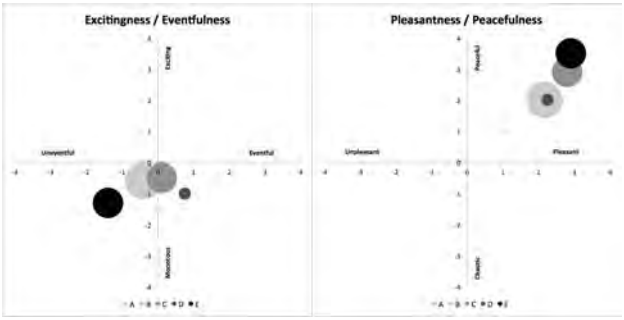


Figure 3: Two graphs showing the results of the evaluation of preferred sounds by the participants. The graphs analyse correlation between excitingness and eventfulness (left) and pleasantness with peacefulness (right), with a high correlation degree for this second case. The physical dimension of the spheres corresponds to the number of people having chosen each sound (preference).

For the criterion excitingness, the preference is less clear but is always presenting a tendency in favour of monotonous sound environments (relative exciting ones).

Finally, regarding the criterion eventfulness, it's even more difficult to describe a clear tendency towards one of the poles; two of the three preferred sounds (E and B) seem to present a narrow tendency towards the uneventful, while C and D are on an average value.

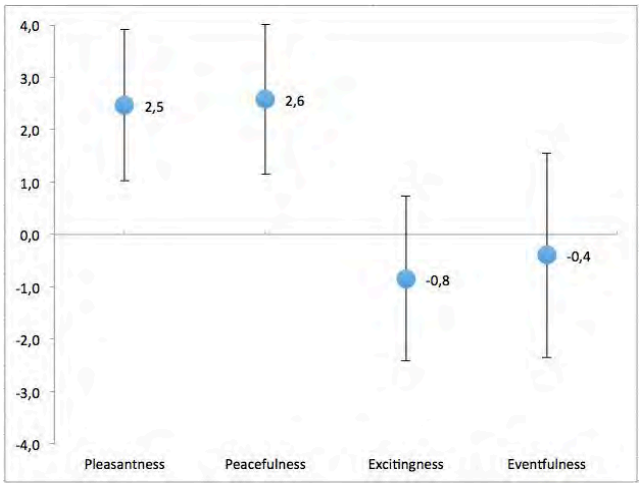


Figure 4: A graph representing tendencies for each one of the four criteria analysed. The graph shows the average value for each criteria as well as the total deviation range.

In general, all criteria are characterized by a low deviation degree of the results for the different sounds, even narrower when just considering the preferred ones (B, C and E). This low deviation degree indicates that the preferred sounds present, for the participants, a quite well-defined character (limited variation in range) regarding the criteria evaluated. Summarising, this character corresponds in general to pleasant and peaceful sounds, with a less marked tendency towards monotony, and a medium degree of eventfulness (not eventful, nor uneventful either).

A participant declared that: "*The sound I listened to most was relaxing, one might say that it brought me back [to focus mode] when my thoughts drifted away. It was just kind of present all the time. Some were disturbing because I thought that they maybe had a too fast tempo or if there was too much happening.*"

A male participant who was critical to his experience stated that the sounds became disturbing/distracting after some time. He described the experience as:*"The sound was too monotonous. After a while it became annoying*."

The majority of the test subjects gave descriptions of their experiences showing that a sound environment should provide both functional and aesthetic values, as illustrated by: "*I prefer the sounds that sounded more musical. I usually listen to music while I work. Other sounds than talking people were pleasant.*" One test

subject who described the preferred sound as restful stated that: "*I felt less disturbed by noise from the office.*"

The participants were asked in open question which was later categorizes, how they chose the sounds. 52.2% answered that they chose the sound because it was the most pleasant, 17.5% felt that the sound they chose was least disturbing, 12.5% answered that the sound helped them to concentrate, 10% answered that the sound blocked out surrounding sounds and 7.5% answered that they didn't know the reason of their choice.

8.1.3 Which sound environments are preferred and used?

Sound B, the most popular sound, was preferred by 31,6% of the participants, 26,3% preferred sound C and just as many preferred sound E, the placebo (26,3% too), 10,5% preferred sound D. The least popular sound was A (dynamically filtered pink noise), which was preferred by only 5,3 % of the participants.

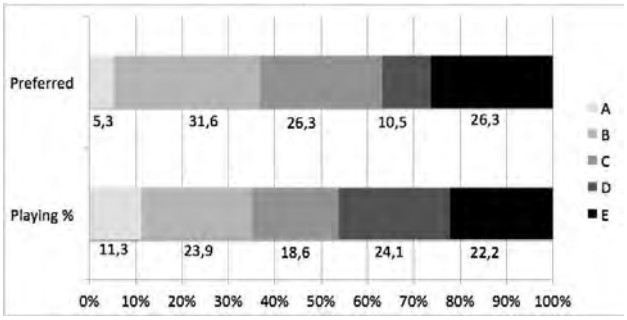


Figure 5. Preferred sounds and playing time percentage.

9. DISCUSSION AND FUTURE WORK

The results provide evidence that the sound bubble presented advantages over the usual acoustic environment in the activity-based office. The large majority of the participants responded that the sound bubble improved the auditory conditions and facilitated focus and concentration. Regarding which of five sound environments that was more attractive and worked better, still remains unclear and needs further investigations. Our results show that the character of the sound should be pleasant and peaceful, but the participants have different opinions of which sound that bring such sensation. One reasonable explanation is that the preference is based on different subjective and individual interpretations of the added sound material; this would point to the problem of sonic/musical taste conditioned by diverse cultural and education backgrounds. In this research project, careful attention was employed on avoiding as far as possible such musical prejudgements by exploring a sonic material as non-musical and abstract as possible, but we cannot discard such tastes operating at different levels other than just as a direct matter of musical language.

We argue that those who experienced the sound environment with the sound bubble as better than without it, actually helped the participants to end up in hearing mode or everyday listening mode. If so, the sound bubble succeeded in establishing a sonic micro-milieu which provided masking, in attention terms, from disturbing background sounds. These results are based on participants' perception of attention and productivity by self-assessment of their experiences only, and should be complemented by studies of actual gain in attention and productivity. That is planned in our future work.

Regarding preferred sounds, the three most popular sequences were one designed for concentration, one for collaboration, and the "placebo" sound. Our findings show that the most popular sounds predominant characteristics were pleasant and peaceful, with a less marked tendency towards monotony and a medium degree of eventfulness; revealing the importance of both aesthetical and functional values.

The least popular sound was the pink noise. These findings correlate with Schlittmeier and Hellbrücks study [21] in which participant ratings spoke in favour of legato music instead of continuous noise as an added acoustic background. They propose that inserted sonic environments must be specially designed with respect to both objective performance effects and subjective ratings. The majority of the participants considered the sound environment in the bubble as positive in comparison to the usual acoustic environment. The reasons given by the participants were that the sound bubble provided aesthetic qualities to the sound environment and supported focused attention by masking out unwanted sounds while still letting the users pick up information from the environment. We can therefore conclude that an active acoustic approach has clear potential for generating place-specific sound environments that better satisfy individual auditory needs in today's office environments.

In terms of future developments of this project, a main concern will guide the next coming phases: enhancing the adaptive nature of the inserted sound environments, developing the system's capacity to "listen to" and analyse the surrounding sounds in order to provide a more accurate sonic response to each situation and context. Further developments should also provide means for the sound bubble to adjust not only to environmental conditions, but also to personal biometric data such as pulse or stress.

It is still not possible to deny the presence of external compositional minds in the development of such inserted environments; however, the final aim of this research project is to evolve this role of a designer/composer more into the one of a sound programmer able to model the patterns of a basic intelligent dialogue with a given context.

10. REFERENCES

[1] Amphoux, P. 2006. L'identite sonore des villes Européennes, CRESSON / IREC, 1993.

[2] Augoyard J.-F. & Torgue H. (eds.) 2006, Sonic Experience. A Guide to Everyday Sounds, McGill-Queen's University Press, Montreal, 216 p.

[3] Atienza R., Billström N. 2012. Fighting "noise" = adding "noise"? Active improvement of high-speed train Sonic Ambiances. In Proceedings of the 2nd International Congress on Ambiances: "Ambiances in action". International Congress on Ambiances, Montreal, Canada, 2012.

[4] Axelsson, Ö, Nilsson, M.E., Berglund, B. 2010. A principal components model of soundscape perception. Journal of the Acoustical Society of America, 128(5), 2836–2846.

[5] Billström N., Atienza R. 2012. Can we improve acoustic environments by adding sound? Internoise 2012, New York, US. Proceedings of the conference. 11p.

[6] Beyer, H., and Holzblatt, K. 1998. Contextual Design: Defining Customer-Centered Systems. Morgan Kauffman, San Francisco.

[7] Chion, M. 1983. Guide des objects sonores. Buchet/Chastel.

[8] De Croon, E.M., Sluiter, J.K., Kuijer, P.P., and Frings-Dresen, M. 2005. The effect of office concepts on worker health and performance: A systematic review of the literature. Ergonomics, 48(2), 119-134.

[9] Eriksson, M. L., and Pareto, L. 2015. Designing Activity-Based and Context-Sensitive Ambient Sound Environments in Open-Plan Offices. *6* (7). http://aisel.aisnet.org/iris2015/7

[10] Franinovic, K., and Visell, Y. 2008. Strategies for sonic interaction design: From context to basic design.

[11] Gaver, W. W. 1993. What in the world do we hear?: An ecological approach to auditory event perception. Ecological psychology, 5(1), 1-29.

[12] Hellström, B. 2012. Acoustic design artefacts and methods for urban soundscapes: a case study on the qualitative dimensions of sounds. InterNoise 2012, New York.

[13] Hellström, B. 2003. Noise Design – Architectural Modelling and the Aesthetics of Urban Acoustic Space, Bo Ejeby Förlag

[14] Hellström, B. 2005. Theories and methods adaptable to acoustic and architectural design of railway stations. In Twelfth International Congress on Sound and Vibration 11-14, Lisbon.

[15] Hermann, T., and Andy H. 2011. The sonification handbook. Logos Verlag, Berlin.

[16] Jahncke, H. 2012. Cognitive Performance and Restoration in Open-Plan Office Noise. Doctoral thesis / Luleå University of Technology, Luleå, Sweden.

[17] Kjellberg, A., Landström, U., Tesarz, M., Söderberg, L. and Åkerlund, E. 1996. The effects of non-physical noise characteristics, ongoing task and noise sensitivity on annoyance and distraction due to noise at work. Journal of Environmental Psychology, 16, 123-136.

[18] Monache, S. D., Polotti, P., and Rocchesso, D. 2010. A toolkit for explorations in sonic interaction design. In Proceedings of the 5th Audio Mostly Conference: A Conference on Interaction with Sound (p. 1). ACM.

[19] Nassiri P, Monazam M, Fouladi Dehaghi B, et al. 2013. The effect of noise on human performance: A clinical trial. The Int. Journal of Occupational and Environmental Medicine, 4, 87-95.

[20] Nilsson M. E. et al. 2010, Auditory masking of wanted and unwanted sounds in a city park, Noise Control Engineering Journal, 58(5), pp. 524-531

[21] Schlittmeier, S. J., and Hellbrück, J. 2009. Background music as noise abatement in open-plan offices: A laboratory study on performance effects and subjective preferences. Applied Cognitive Psychology, 23(5), 684-697.

[22] Smith, S. E., Stephan, K. L., and Parker, S. P. 2004. Auditory warnings in the military cockpit: A preliminary evaluation of potential sound types (No. DSTO-TR-1615). Defence Science and Technology Organisation Edinburgh (Australia) Air Operations Div.

[23] Västfjäll, D. 2002. Influences of current mood and noise sensitivity on judgments of noise annoyance. The Journal of Psychology: Interdisciplinary and Applied, 136, 357-370.

[24] Wang, F., and Hannafin, M. J. 2005. Design-based research and technology-enhanced learning environments. Educational Technology Research and Development, 53(4), 5-23.

"TREES": AN ARTISTIC-SCIENTIFIC OBSERVATION SYSTEM

Marcus Maeder
Zurich University of the Arts
ZHdK, Institute for Computer
Music and Sound Technology ICST
marcus.maeder@zhdk.ch

Roman Zweifel
Swiss Federal Institute for
Forest, Snow and Landscape
Research WSL
roman.zweifel@wsl.ch

ABSTRACT

In our research project *«trees: Rendering eco-physiological processes audible»* we connected acoustic emissions of plants with ecophysiological processes and rendered audible natural phenomena that aren't normally noticeable in an artistic way. The acoustic emissions of a tree in the Swiss Alps were recorded with special sensors, and all other non-auditory ecophysiological measurement data (e.g. the trunk and branch diameters that change depending on water content, the sap flow rate in the branches, the water present in the soil, air moisture, solar radiation, etc.) were sonified, i.e. translated into sounds. The recordings and sonified measurements were implemented in a number of different media art installations, which at the same time served as a research environment, in order to examine and experiment artistically with the temporal and spatial connections between plant sounds, physiological processes and environmental conditions in an artistic-scientific observation system.

1. INTRODUCTION

The link between trees and various climatic processes is usually not immediately apparent. Trees and plants do not live merely on moisture from rain, sunlight (which drives gas exchange) and nutrients from the soil: they absorb carbon dioxide from the air and produce the oxygen that we breathe, maintaining our climate and biosphere [1]. Gathering ecophysiological data by measuring the local climatic and environmental variables and the physiological processes within a plant in response to changes in these variables has become an important method to investigate the relationship between climate change and vegetation dynamics [2]. It helps to determine physiological thresholds of plants in terms of increasing temperature and consequently drought stress.

The Scots pine (*Pinus sylvestris*) in Valais, a valley in the inner Swiss Alps, has experienced high mortality rates for some decades now: this phenomenon is believed to be caused *inter alia* by the effects of climate change, e.g. longer drought periods [3]. A downy oak (*Quercus pubescens*), for example, is better able to withstand the current climatic conditions whereas the Scots pine is pushed beyond its physiological limits despite the fact that both tree species have coexisted there

Figure 1. Scots pine forest in Salgesch/VS.

for thousands of years [4]. Consequently, a shift in the abundance of tree species is observed [5]. The ecophysiological knowledge acquired is used to explain the underlying processes: Hence the interest in cooperation between a biologist and an artist to study and depict the complex relationship between tree physiology and the climate on one hand and to explore the possibilities of acoustic and artistic representations of ecophysiological processes in trees on the other. Rendering physiological processes audible (e.g. water transport or trunk diameter changes) allows us to identify and better understand plants' responses to climatic processes.

2. ACOUSTIC EMISSIONS OF PLANTS AND DROUGHT STRESS

Plants emit sounds – a bigger part of these sounds are of transpiratory/hydraulic origin and are therefore related to

the circulation of water and air within the plant as part of the transpiration process [6]. The frequencies of the acoustic emissions lie mostly in the ultrasonic range (UAE = ultrasonic acoustic emissions), depending on the species-specific characteristics of the plant tissue [7]. The loudest UAE that occur in a plant arise due to drought stress. The excessive water tension in the water-conducting system leads to a collapse of single water columns in the plant vessels. They embolize, and this event releases an impulse that travels as an elastic wave through the tissue and may be detected as an UAE on the plant surface [8].

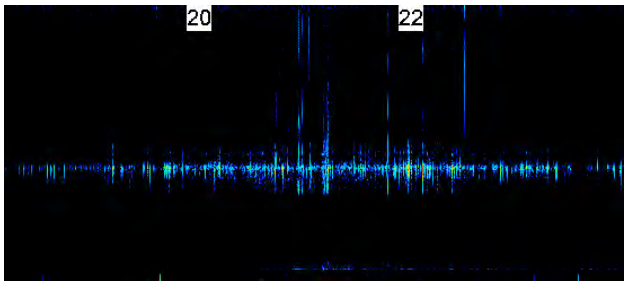


Figure 2. Sonagram of a Scots Pine: Ultrasonic acoustic emissions (UAE) appear as vertical spikes with different extents.

Thirsty and stressed plants make an inaudible noise [9]. UAE from plants lead to conclusions on their state and on the environmental conditions. Measurements of the amount and temporal occurrence of UAE are being used in ecophysiology to assess a plant's vulnerability to drought stress. More and longer occurring UAE in a plant are an indicator of increasing stress and desiccation. During our research project it became clear that our observation system could make a fundamental phenomenon tangible: namely, how plants react to ever-longer periods of heat and drought in the course of climate change.

When growth begins in spring, the tree benefits for a while from the water reserves from the winter that are still in the soil. However, the many acoustic emissions that occur already in spring (during the growth period) reveal a situation of immense stress in which the tree exists, because it must maintain the necessary turgor pressure in the cells in order to grow. If the water reserves in the soil have been exhausted, and there has been no precipitation in the meantime, the tree reacts by restricting its transpiration and growth (indicated by a vanishing of the UAE), in order to protect itself from dehydration. If periods of heat and drought become ever-longer as a result of climate change, trees will become susceptible to diseases and parasite infection – and will die early.

The acoustical measurement equipment we used for detecting and measuring the UAE consisted of a combination of UAE sensors from non-destructive testing (Vallen Systems VS150-M) with devices from bioacoustical research, the field of investigation of acoustic behavior of animals (modified hydrophone preamplifiers, high sampling rate audio interfaces and adapted measuring/recording software, all by Avisoft). We have built also our own acoustic sensors, based on DIY technology like those used in many artistic experiments and performances: a copper wire pin was soldered onto a piezo element, in order to be able to couple them optimally into the tissue of a plant (Fig. 3).

Figure 3. Self-built piezo needle sensor.

3. DATA SONIFICATION

The representation of data using sound (among other means) can help to exploit the effectiveness of our sense of hearing in grasping complex contexts both through immediate orientation in space and intuitive classification of sound characteristics [10]. Sonification offers a deep and broad insight into multidimensional data, enabling us to recognize patterns and providing an aesthetic and emotional experience of scientific discoveries. In contrast, many data sonification experiments in the scientific field are characterized by their shortcomings for inexperienced listeners. Very little weight is usually given to aesthetic judgments in the generation and use of sounds in scientific data sonification. What is mostly sought is simply a clear differentiation between individual sounds, which means that the artistic design of sonifications is usually rudimentary. Our aim was to increase the accessibility of our data sonification insofar as that we set it up from the very outset in accordance with musical criteria, i.e. the biological connections and correlations corresponded with the harmony, dramaturgy and emphasis of the artistically arranged sounds. By these

means a generative piece of music was created that is controlled by the data flows used, and which gathers together the individual processes and phenomena in a holistic musical experience.

The recorded plant signals were transposed into the audible range (audification) [11] and are played as an audio sample in the sonification system. The sounds of the weather are also played as a sample (wind and rain). All non-auditory measurement data have been sonified, i.e. measurement data series of individual physiological processes control the characteristics of individual sounds (parameter mapping) [12].

The different sonification modules are implemented in a Max/MSP patch, which replays measurement data from spring until summer 2015. For an adequate (temporal) experience of the most important processes, the speed of the running system is increased up to 36 times the normal speed, i.e. the 10-minute measuring intervals. Environmental data is mapped on the outer side of the spatial audio system of the *trees: Pinus sylvestris* installation (spatial audio versions), while tree data is played back on single speakers of the system, according to the spatial position and geographic orientation of the sensors on the plant.

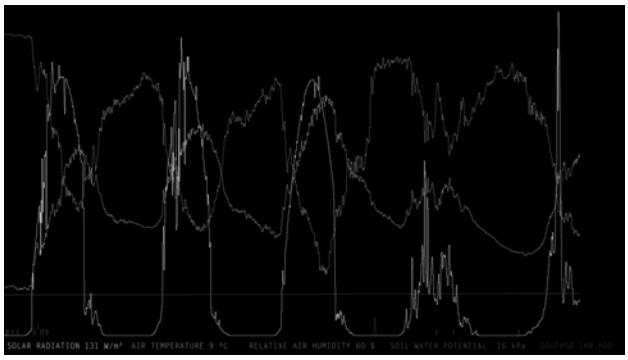


Figure 4. Data display of the sonification module of *trees: Pinus sylvestris.*

The sounds that we used to sonify the measurement data can be divided into two groups: field recordings (rain, wind and plant sounds) and synthetic sounds. A larger number of phenomena do not manifest themselves acoustically, and we created metaphorical sounds that portrayed a single phenomenon, for instance sunlight or air humidity, in the best way. For sunlight, for instance, we took a string-like sound; for air humidity, a transformed and filtered burbling sound of a creek. All sound sources have a specific static or dynamic location within the audio system. The sun sound moves across the firmament, from east to west; wind comes from varying directions, louder if strong, softer if weak. As the sun rises, transpiration within the tree starts, and sap flow noises become louder. When the diurnal course reaches its peak at noon, cavitation pulses start, later decreasing during the course of the afternoon. The following table shows the measurement data, the sound characters that we associated with them and how they have been spatialized within the audio system (Tab. 1):

Measured data	**Sound character**	**Playback parameters**
Daylight [RGB brightness]	Atmospheric synthetic sound	Amplitude, controlled by video brightness
Solar radiation [W/m^2]	String-like, synthetic sound	Amplitude
Sun position [azimuth, elevation]	Same	Spatial position
Air temperature [°C]	-	Main volume
Rel. air humidity [%]	Water-like, synthetic sound	Pitch, amplitude
Rain [mm]	Field rec.: Rain	Amplitude, spatial position
Wind [m/sec., azimuth]	Field rec.: Wind	Amplitude, spatial position
Soil water potential [kPa]	Field rec.: Seeping water	Amplitude, placed on discretely driven speakers near ground
Tree branch diameter [µm]	-	High pass filter, applied on sap flow sound
Tree sap flow [g H_20/h]	Floating water, transposed up and filtered	Amplitude, placed on discretely driven speakers
Tree UAE	Field rec.: UAE	Pitch, amplitude, placed on discretely driven speakers

Table 1. Measurements and assigned sounds.

4. TREES: PINUS SYLVESTRIS

After a first prototype [13], we developed four versions of our artistic-scientific observation system *trees: Pinus sylvestris* until now. The *stereo/IP cam version* for two speakers and/or three headphones and three TFT monitors, the larger *spatial audio installation* (Fig. 5) and the adaption for ICST's *Immersive Lab* (Fig. 6) [14], as well as the adaption for ICST's *FlowSpace* [15]. The same sonification algorithms are implemented in all versions but are differently mapped on the speakers/headphones and present different video footage. In the stereo version, we used video footage from two IP cams on the stem of our measurement tree, each focusing on a branch of the plant.

The spatial audio system consists of an octagon carrying 36 self-built omni-directional speakers. It is designed as an accessible three-dimensional speaker array, where virtual sound sources are moved and placed within a defined space, and listeners can walk around inside the

system. The speaker matrix that we have developed is a hybrid sound system:

Figure 5. Spatial audio version of *trees: Pinus sylvestris.*

An Ambisonics [16] sound field is mapped onto the tube matrix, but some of the speakers are driven discretely. A 24" touch screen at the centre makes the installation an explorative, self-explanatory artistic system: The visitor is able to switch sound sources on and off to identify individual phenomena and their sonifications (see Fig. 4). A time-lapse video of the tree and its surroundings informs the visitor visually about the local climatic conditions (weather, time of day and light intensity). These images were taken by a so-called tree canopy camera, i.e. a fish-eye camera system that biologists use for measuring the light intensity coming through a canopy to investigate a tree's state of health or the light exposure of plants growing on the forest floor for instance. Furthermore, visitors may observe a graphic representation of the local climatic measurements and the correlations between the individual phenomena as well as a graphic display of the peak frequencies of the measured acoustic emissions at different locations along the plant's trunk and branches.

Figure 6. *trees: Pinus sylvestris* for the *Immersive Lab.*

The adaption for ICST's *Immersive Lab* (Fig. 6) marks an important shift regarding immersion. The projection surfaces of the system are touch sensitive: This enables the visitor to touch single branches of the tree, hear the branch-related acoustic emissions immediately and see a graphical representation of the local measurement data.

5. CONCLUSIONS

Our research project demanded of each participant to engage deeply with the mindset of the other disciplines involved in the project, in order to be able to develop a scientifically usable and artistically appropriate system that would make the key processes behind plant sounds identifiable and tangible. Aesthetic questions therefore already determined the structure of the experiment, for example where and how the cameras should be installed on the tree, or how the ecophysiological sensors should be mounted in order to enable an adequate and comprehensive presentation of the life processes and environmental conditions in an immersive environment. On the artistic side, consideration had to be made of the scientific and technological conditions; beyond this, the core artistic task was to develop adequate sonic representations of the non-auditory ecophysiological processes.

The reconstruction and staging of the life processes and environmental conditions of a tree in an artistic-technical environment has led to a completely new field of research and design for all those involved, with an innovative instrument: correlations of measured values and patterns in natural processes become aesthetic effects – abstract measurement data are reflected in images and sounds. The image of nature produced with digital technology demands an artistic nuancing of the acoustic and visual presentations, so that, for example, the variety of sounds present in the system do not disturb or overlap each other. Therefore, data must be interpolated and filtered, in order to be able to experience individual processes.

Another important point in our project emerged ever more clearly during the research work. The project *trees* dealt with the production of a new form of holistic knowledge that is not conveyed merely via the verbalisation of findings in a research report, but rather in a directly tangible auditory and visual (medial) form. The intention of the implementation of our artistic-scientific observation system was to create an all-encompassing experience from very different and complex data sets, and thus to draw a holistic picture of the life processes and environmental conditions of a tree that is under pressure from changing climatic conditions. The balance between the knowledge and practices applied is key – the artistic imagination of scientific objects must receive the same attention as the scientific foundation of the aesthetic objects.

The great success of the project in the media and among political representatives is due to the fact that we have

managed, in an artistic-scientific manner, to make it possible to grasp processes that are not usually noticeable and thus create a multifaceted, direct and comprehensive experience of natural phenomena. Our image of nature, in particular our perception of the plant kingdom, is still dominated by a perspective that treats life processes like the mechanistic functions of inanimate objects. Yet the animate object often reveals itself only by means of a change in perspective, a reduction in distance and the suspension of differences (between human subjects and natural objects). Present-day media technologies place us in a position to experience and interpret nature and natural objects and processes anew, in an immersive situation.

6. ACKNOWLEDGEMENTS

The research project *trees: Rendering Ecophysiological Processes Audible* has been funded by the Swiss National Science Foundation (SNSF) (100016_143958/1), the Zurich University of the Arts ZHdK and the Swiss Federal Institute for Forest, Snow and Landscape Research WSL.

7. REFERENCES

[1] Larcher, W. 2003. Physiological Plant Ecology - Ecophysiology and Stress Physiology of Functional Groups. Berlin: Springer.

[2] Teskey, R., Wertin, T., Bauweraerts, I., Ameye, M., McGuire, M. A., Steppe, K. 2015. Responses of tree species to heat waves and extreme heat events. Plant Cell and Environment 38(9): 1699-1712.

[3] Rigling, A., Bigler, C., Eilmann, B., Feldmeyer-Christe, E., Gimmi, U., Ginzler, C., Graf, U., Mayer, P., Vacchiano, G., Weber, P., Wohlgemuth, T., Zweifel, R., Dobbertin, M. 2013. Driving factors of a vegetation shift from Scots pine to pubescent oak in dry Alpine forests. Global Change Biology 19(1): 229-240.

[4] Zweifel, R., Steppe, K., Sterck, F. J. 2007. Stomatal regulation by microclimate and tree water relations: interpreting ecophysiological field data with a hydraulic plant model. Journal of Experimental Botany 58(8): 2113-2131.

[5] Zweifel, R., Rigling, A., Dobbertin, M. 2009. Species-specific stomatal response of trees to drought – a link to vegetation dynamics. Journal of Vegetation Science 20: 442-454.

[6] Milburn, J. A. and Johnson, R. P. C. 1966. The conduction of sap. II. Detection of vibrations produced by sap cavitation in Ricinus xylem. Planta (Berl.) 69: 43-52.

[7] Kawamoto, S. and Williams R. S. 2002. Acoustic Emission and Acousto-Ultrasonic Techniques for Wood and Wood-Based Composites. United States Department of Agriculture, Forest Service – General Technical Report FPL-GTR-134.

[8] Jackson, G. E. and Grace, J. 1996. Field measurements of xylem cavitation: are acoustic emissions useful? Journal of Experimental Botany, Vol. 47, No. 304: 1643-1650.

[9] Zweifel, R., Zeugin, F. 2008. Ultrasonic acoustic emissions in drought-stressed trees - more than signals from cavitation? New Phytologist 179: 1070-1079.

[10] Hermann, T., Hunt, A., Neuhoff, J. G. (Ed.) 2011. The Sonification Handbook, Berlin: Logos.

[11] Cramer, G. 1994. Auditory Display: Sonification, Audification and Auditory Interfaces. Proceedings Volume 18, Santa Fe Institute Studies in the Sciences of Complexity (Book 18).

[12] Horiguchi, Y., Miyajima, K., Nakanishi, H., Sawaragi, T. 2013. Parameter-Mapping Sonification for Manual Control Task: Timbre and Intensity Manipulation to Sonify Dynamic System State. Proceedings SICE Annual Conference 2013, Nagoya University, Nagoya, Japan: 2341-2346.

[13] Maeder, M. and Zweifel, R. 2013. Downy Oak: Rendering Ecophysiological Processes In Plants Audible. Proceedings of the Sound and Music Computing Conference 2013, SMC 2013, Stockholm, Sweden: 142-145.

[14] http://immersivelab.zhdk.ch from 10.04.2016.

[15] Bisig, D., Schacher, J. C. 2011. Flowspace – A Hybrid Ecosystem, Proceedings of the International Conference on New Interfaces for Musical Expression, Oslo, Norway, 2011.

[16] Schacher, J. C. 2010. Seven Years Of ICST Ambisonics Tools For MaxMSP, Proceedings of the 2nd International Symposium on Ambisonics and Spherical Acoustics May 6-7, Paris, France, 2010.

$VISA^3$: REFINING THE VOICE INTEGRATION/SEGREGATION ALGORITHM

Dimos Makris
Dept. of Informatics,
Ionian University,
Greece
c12makr@ionio.gr

Ioannis Karydis
Dept. of Informatics,
Ionian University,
Greece
karydis@ionio.gr

Emilios Cambouropoulos
Dept. of Musical Studies,
Aristotle University of
Thessaloniki, Greece
emilios@mus.auth.gr

ABSTRACT

Human music listeners are capable of identifying multiple 'voices' in musical content. This capability of grouping notes of polyphonic musical content into entities is of great importance for numerous processes of the Music Information Research domain, most notably for the better understanding of the underlying musical content's score. Accordingly, we present the $VISA^3$ algorithm, a refinement of the family of VISA algorithms for integration/segregation of voice/streams focusing on musical streams. $VISA^3$ builds upon its previous editions by introduction of new characteristics that adhere to previously unused general perceptual principles, address assignment errors that accumulate affecting the precision and tackle more generic musical content. Moreover, a new small dataset with human-expert ground-truth quantised symbolic data annotation is utilised. Experimental results indicate the significant performance amelioration the proposed algorithm achieves in relation to its predecessors. The increase in precision is evident for both the dataset of the previous editions as well as for a new dataset that includes musical content with characteristics such that of non-parallel motion that are common and have not yet been examined.

1. INTRODUCTION

It is a common understanding of music listeners that musical content can be separated to multiple 'voices'. Nevertheless, it is widely accepted [1–3] that the notion of a 'voice' is far from well-defined as it features in a plethora of alternative meanings, especially when polyphonic and homophonic elements are included.

In most occasions, the term 'voice' refers to a monophonic sequence of successive non-overlapping musical tones, as a single voice is assumed not to contain multi-tone sonorities. In some cases though, provided that 'voice' is examined in the light of auditory streaming, it is possible that the standard meaning is insufficient. In these cases, a single monophonic sequence may be perceived as more than one voices/streams (e.g., pseudopolyphony or implied polyphony) while a sequence containing concurrent notes may be perceived as a single perceptual entity (e.g., homophonic passages) [4].

Musical auditory stream integration/segregation defines how successions of musical events are perceived to be coherent sequences and, at the same time, segregated from other independent musical sequences. A number of general perceptual principles govern the way musical events are grouped together in musical streams [1, 2].

Given the ambiguity of 'voice' segregation definition, the process can be separated into two different broad categories based mostly on whether the resulting voices are monophonic or not. The scenario wherein the resulting voices of the segregation are monophonic is titled as 'voice segregation'. On the other hand, when the resulting segments are organised in perceptually coherent groups that may include overlapping notes, then the process is referred to as 'stream segregation'. Accordingly, this work's focal point lies on stream segregation based on quantised symbolic data.

Musical content's voice/stream segregation is of great importance to Music Information Research (MIR) as it allows for efficient and higher quality analytic results, such as the identification of multiple voices and/or musical streams for the purpose of processing within the voices (rather than across voices) [2]. All in all, voice and stream segregation approaches aim at grouping notes of polyphonic musical content into entities that allow for better understanding of the underlying musical content's score [5], and for this are essential to MIR.

1.1 Motivation and Contribution

Existing methodologies of stream segregation, as extensively described in Section 2, do not utilise as many as possible of the general perceptual principles [2] that govern the way musical events are grouped together in musical streams. Moreover, previous implementations usually present low precision due to erroneous early stream assignment propagation until the end of the piece. In addition, most works of voice/stream segregation focus solely on a genre/type of musical content, thus providing genre-customised experimentation. One further setback of this genre-customised experimentation is the lack of breadth of available ground-truth for further algorithms' examination.

Accordingly, the contribution of this work is summarised as follows:

- Incorporates the general perceptual principle of *Co-*

Modulation Principle that allows for ameliorated vertical integration.

- Proposes a methodology that segments musical pieces into grouping entities that allow for revision and elimination of the initial error propagation phenomenon.
- Extends the available stream segregation domain datasets with ground truth by providing new, non-pop, human-expert produced annotation of streams in musical pieces.

The rest of the paper is organised as follows: Section 2 describes background and related work and Section 3 provides a complete account of the proposed method. Subsequently, Section 4 presents and discusses the experimentation and results obtained, while the paper is concluded in Section 5.

2. RELATED WORK

Research on computational modelling of segregation of polyphonic music into separate 'voices' has lately received increased attention, though in most of these cases, 'voice' is assumed to be a monophonic sequence of successive non-overlapping musical tones.

The work of Temperley [6] proposes a set of preference rules aiming at avoiding large leaps and rests in streams, while minimising at the number of streams, avoiding the common tones shared between voices and minimising the fragmentation of the top voice. In [7], Cambouropoulos makes the case for tones being maximally proximal within streams in temporal and pitch terms, the minimisation of the number of voices and the lack of streams' crossing, i.e. the maximum number of streams to be equal to the number of notes in the largest chord. Chew and Wu [8] propose an algorithm based on the assumption that tones in the same voice should be contiguous and proximal in pitch, while voice-crossing should be avoided, i.e. the maximum number of voices to be equal to the number of notes in the largest chord. Szeto and Wong [9] present stream segregation employing a clustering modelling technique. The key assumption therein is that a stream is to be considered as a cluster since it is a group of events sharing similar pitch and time attributes (i.e. proximal in the temporal and pitch dimensions). Their algorithm determines automatically the number of streams/clusters. As aforementioned, all of these voice separation algorithms assume that a 'voice' is a monophonic succession of tones, thus focusing on the voice separation scenario.

The work by Kilian and Hoos [10] differs from the voice separation scenario as it allows for entire chords to be assigned to a single voice. Accordingly, more than one synchronous notes can potentially be assigned to one stream. Their solution segments the piece into slices with each slice containing at least two non-overlapping notes. Penalty values are used in an aggregating cost function for features that promote segregation such as large pitch intervals, rests / gaps, note overlap between successive notes, large pitch intervals and onset asynchrony within chords. The notes of each slice are separated into streams by minimisation of the cost function. The penalty values are user-adjustable in order lead to a different separation scenarios of voices by testing alternative segregation options. The maximum number of voices is again user-defined or automatically selected based on the number of notes in the largest chord. The pioneering aspect of the proposal of Kilian and Hoos lies on the fact that multi-note sonorities within single voices are allowed. Accordingly, their algorithm has a different scope/target, i.e. to split notes in different staves on a score. It takes perceptual principles in account but the result is not necessarily perceptually meaningful.

As far as the evaluation of voice/stream separation algorithms is concerned, in most of the aforementioned works, it has been performed solely on classical musical pieces. Guiomard-Kagan et. al [5] expanded their corpus to evaluate most existing voice and stream separation algorithms by adding 97 popular music pieces containing actual polyphonic information. However, the annotation used therein was based on ground truth created with monophonic voices and not streams, and thus is not applicable to our proposal.

2.1 The VISA Algorithm

The previous editions of the Voice Integration/Segregation Algorithm VISA algorithm proposed originally by Karydis et al. [11] and extended by Rafailidis et al. [3] are all based on the perceptual principles for stream separation as proposed by Bregman [12]. Basic perceptual principles, such as grouping rules based on similarity and proximity (i.e. proximal or similar entities in terms of time, space, pitch, dynamics, timbre are to be interrelated in perceptually-valid groups), have been employed in the last decades for modeling music cognition processes [13]. Huron [14] maintains that the main purpose of voice-leading in common practice harmony is to create perceptually independent musical lines/voices and presents a set of 10 perceptual principles that explain a large number of well-established voice-leading rules. The edition of the VISA algorithm proposed herein, draws on the perceptual principles presented by Huron with alterations as proposed by Cambouropoulos in [2]. The principles that are used in the previous implementations of the VISA algorithm are:

1. *Synchronous Note Principle*: Notes with synchronous onsets and same IOIs (durations) tend to be merged into a single sonority [11].
2. *Principle of Temporal Continuity*: Continuous or recurring rather than brief or intermittent sound sources' evoke strong auditory streams [14].
3. *Pitch Proximity Principle*: The coherence of an auditory stream is maintained by close pitch proximity in successive tones within the stream [14].

In order to make the distinction more clear, the original edition of the VISA algorithm as proposed by Karydis et al. in [11] is henceforth referred to as VISA07 while the edition prposed by Rafailidis et al. in [3] is denoted as VISA09.

2.1.1 Previous Editions of VISA

All editions of the VISA algorithm receive as input the musical piece in the form of a list L of notes that are sorted

according to their onset times, a window size w, and a threshold T. The output is the number V of detected musical streams. Notice that none of the VISAs demand an a-priori knowledge of the number of voices.

In detail, VISA07 and VISA09 moved in a step-wise fashion through the input sequence of musical events. The set of notes having onsets equal to the position of a "sweep line" was denoted as Sweep Line Set (SLS). Then, every SLS was divided into clusters by partitioning the notes into a set of clusters C. The clustering procedure was achieved according to the *Synchronous Note Principle*. For a set of concurrent notes at a given SLS, it had to be determined how to merge these on the set of clusters C. Since it is possible that synchronous notes may belong to different streams, VISAs examined the musical context w around these. If inside the context window, most co-sounding notes had the same onsets and offsets, implying thus a homophonic texture, then these were merged. Otherwise, this being most possibly a polyphonic texture, the notes were not merged in single sonorities. In addition, as notes with different offsets produce different clusters, each SLS was split into a number of note clusters.

In VISA07, the cluster separation was following only the *Synchronous Note Principle* while in VISA09 the *Break Cluster* module was introduced as an extra method for vertical integration. In this case, for every SLS, if the texture is homophonic and all notes have the same duration, this procedure looked ahead in the next three SLSs; if there existed more clusters in one of the following SLSs, VISA09 moved backwards and broke one by one its preceding clusters, according to the *Pitch Proximity Principle* until the current SLS cluster was examined.

Given the set of clusters C for every SLS, the horizontal streaming principle (i.e. the combination of *Temporal Continuity* and *Pitch Proximity* principles) was used to break these down into separate streams. For each SLS in the piece, a bipartite graph was formed in order to assign these to streams where one set of vertices corresponded to the currently detected streams (V) and the other set corresponded to the clusters in C. The corresponding edges represented the cost for each assignment. The cost function calculated the cost of assigning each cluster to each voice according to the *Temporal Continuity Principle* and the *Pitch Proximity Principle*.

Moreover, VISA09 included a procedure that forced the algorithm to switch onto two streams when the texture is homophonic. This was done in order not keep 'alive' extra streams (e.g. a third or fourth stream) given that the tendency was to have one or two constant streams (melody and harmonic accompaniment).

Then, using a dynamic programming technique, the best matching (lowest cost) was found between previous streams and current clusters. Finally, two additional constraints were taken into account: the former enforced stream crossing not to be allowed while the latter ensured that the top stream should be minimally fragmented [6].

2.1.2 Problems of VISA

VISA09 was tested on several musical examples that were carefully selected so as to contain a constant and small number of (up to three) streams. Most of these are homophonic pieces and the algorithm performed well in terms of precision since procedures were implemented to support better homophonic stream assignment. However, further examination showed that the algorithm's precision was diminished when tested on different music styles that contained non-homorhythmic homophonic accompanimental textures with more than 2 streams. The same phenomenon can be seen in pieces of the dataset in [3] with such homophonic texture but containing more than two streams, wherein the algorithm failed to produce a proper separation. Moreover, VISA09 was not designed to detect potential non parallel movement between notes with same onsets and offsets. Figure 1 shows an example of a non-classical piece containing non-parallel movement between notes wherein VISA09 tends to create single cluster sonorities due the homophonic texture leading to wrong stream assignment.

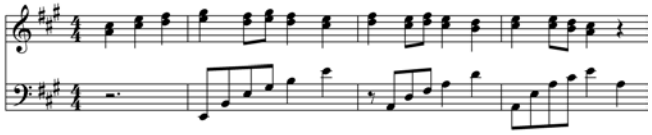


Figure 1. Excerpt from the couplet of the Greek folk song *Kaith Xwmata - Ki an se agapw den se orizw*.

In addition, the horizontal stream assignment moving by SLS from the beginning of a piece until the end can be problematic in certain cases, as the cost calculation in every SLS for assigning the streams on the current clusters is based on principles and costs of previous assignments. Therefore, if the algorithm detects in previous SLSs a wrong number of streams or clusters, it will possibly continue to accumulate wrong calculations for all the remaining SLSs even though that the piece could be very simple as far as stream assignment is concerned. This scenario was observed mainly in pieces that contain three or more streams.

Finally, the choices of the *Break Cluster* approach and the homophonic detection, which force the algorithm to switch back to the two basic streams, seem very specialised for certain (genres of) musical pieces, especially given that research for voice/stream separation has thus far mainly focused on classical music pieces.

3. THE PROPOSED METHOD

The proposed revision of VISA, the $VISA^3$ edition differs from the previous two, not only in functionality, but by additionally performing a step further after vertical integration as well as having been tested on popular music too, in addition to the common dataset of the previous two versions of VISA. We propose the use of the *Co-Modulation Principle* for further vertical integration and a customised *Contig Segmentation* approach, based on the work of Chew and Wu [8] using $clusters$. Figure 2 presents the steps of our revision which are:

1. Vertical Integration: Merging Notes into Single Sono-

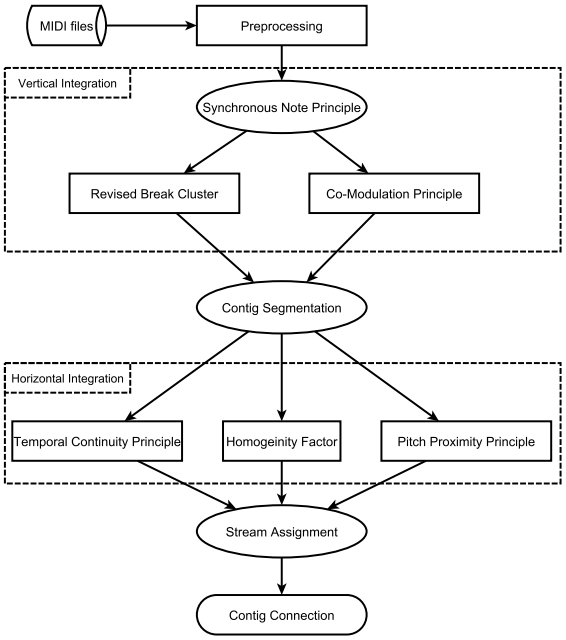


Figure 2. The $VISA^3$ algorithm.

rities using the *Synchronous Note Principle* and then examining special cases for further integration with the *Break Cluster* technique and the *Co-Modulation Principle*.

2. Contig Segmentation: Segmentation of the piece into contigs from the previous step.
3. Horizontal Integration: Stream matching within contigs using horizontal streaming principles and other factors such as homogeneity.
4. Contig Connection: Integration of contigs by connecting their streams on the segmentation boundaries.

3.1 Merging Notes into Single Sonorities

$VISA^3$ accepts as input the musical piece (i.e. a quantised MIDI file) in the form of a list L of notes that are sorted according to their onset times, a window size w and the homophony threshold T, exactly the same parameters as the previous editions of the VISA algorithm. After merging the notes into clusters according to the *Synchronous Note Principle*, further vertical integration takes place with the new revised *Break Cluster* module and the *Pitch Co-modulation Principle*.

3.1.1 Break Cluster Module

The Break Cluster module is activated when the local context is mostly homophonic and a number of notes are integrated vertically, producing thus a cluster in the current SLS. The following two significant changes occur in relation to the previous versions of the VISA algorithm:

1. Instead of looking ahead in the next three SLSs, the revised procedure of $VISA^3$ looks for the following SLSs that appear within a window size w,

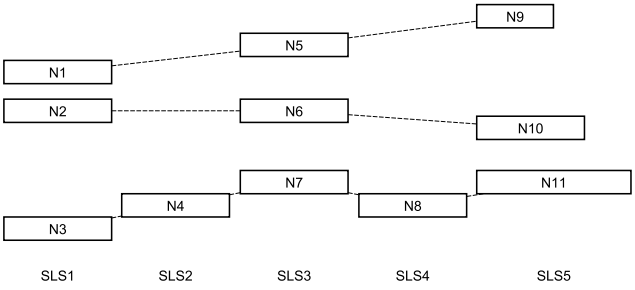


Figure 3. Breaking vertical clusters. Vertical clusters in SLS_1 and SLS_3 are broken retrospectively as the last SLS_5 comprises of three clusters; thus, this extract is separated into three streams.

2. In VISA09 the look-ahead procedure works only for single large clusters with the same number of streams, then ceases to function if it identifies more on the subsequent SLSs and starts breaking these according to pitch proximity. In $VISA^3$, the procedure doesn't stop in cases where the next SLS has less streams than the initial cluster, but it skips it and continues with the following until it finds the breaking point. In this way, the clusters that are not necessarily consecutive are being examined.

Figure 3 shows an example where the notes are in single clusters and the context is homophonic. All notes in SLS_1 are clustered vertically into a single cluster. Therefore the Break Cluster procedure is activated and looks the next SLSs in a window size w. It skips SLS_2 and SLS_4 as it detects fewer streams than SLS_1 and stops on SLS_5 as it finds three clusters: {N9}, {N10} & {N11}. Moving backwards, the process breaks SLS_3 and SLS_1 to {N5}, {N6}, {N7} & {N1}, {N2}, {N3}, respectively, based on the *Pitch Proximity Principle*. It is worth noting that if the process finds clusters with more voices than SLS_1, all combinations will be checked.

3.1.2 Pitch Co-modulation Principle

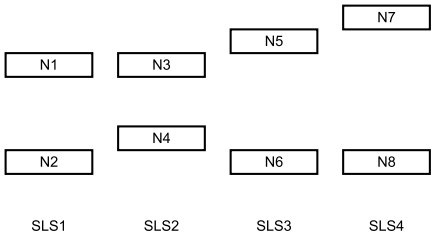
$VISA^3$ features a functionality aiming at detecting non-parallel movement between voices of consecutive vertically integrated clusters which the *Synchronous Note Principle* cannot separate. This principle is based on Huron's *Pitch Co-modulation Principle* [14]: "The perceptual union of concurrent tones is encouraged when pitch motions are positively correlated".

The procedure works as follows: In every SLS in which clusters with two or more notes are detected, it looks ahead up to a window of size w and attempts to create monophonic chains within consecutive clusters of the same number of notes. It examines whether two chains follow the same overall direction (i.e. if the notes move in parallel or not) by calculating the deviation in the pitch differences between the corresponding chain notes. Accordingly, there are two cases to be examined: two note chains in two-note cluster sequences and constant three or more note chains in three of more note clusters.

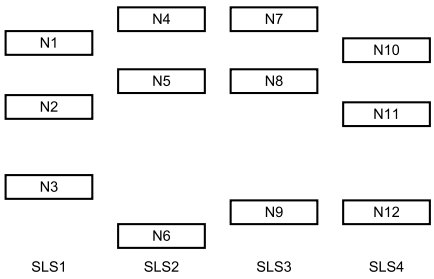
As far as the first case is concerned, the distinguishing

task is rather clear: if the concurrent notes within a chain move in non-parallel direction, these are separated and the procedure moves backwards breaking, in every SLS, the corresponding cluster into two separate clusters following the technique found in [15]. For the latter case, i.e., for larger clusters, each such cluster is separated into a set number of note chains. If the direction of notes between two chains is the same (i.e. parallel movement) then the notes of the two chains remain in the same stream. Else, if the direction of notes is different, then these form different streams. On the other hand, if there is no correlation between the movement of each stream within the chain then the cluster is separated.

The proposed methodology is based on the following two assumptions: First, the number of notes of the consecutive large clusters has to be constant. Otherwise, a cluster chain is terminated when clusters with more or less notes are found. Secondly, the direction of notes refers to the contrapuntal motion between two melodic lines [16]. While in cluster chains with two notes we seek for *parallel* motion, in this case we seek for *similar* motion, where the notion of similar motion refers to motion in the same direction. Thus, both chains move up or down but the interval between these is different in every SLS. Figure 4(a) presents examples of both cases where the notes inside the chains move in non-parallel direction and thus require separation. In Figure 4(b), the upper two streams move in parallel and thus do not require separation, in contrast to the third (lower) stream.



(a) Cluster chains with two voices



(b) Cluster chains with three or more voices

Figure 4. Examples of non-parallel movement on consecutive vertically integrated clusters.

3.2 Contig Clustering Process

The Contig Clustering process is based on the work of Chew and Wu [8] that proposed a "contig map" for voice separation. A *contig* is a collection of sequences of successive notes that belong to the same voice and the overlap depth (number of note sequences) at any time is constant. In the context of $VISA^3$, the *contig* clustering process segments a piece into contigs according to stream count and then reconnects the fragments in adjacent contigs using a distance strategy.

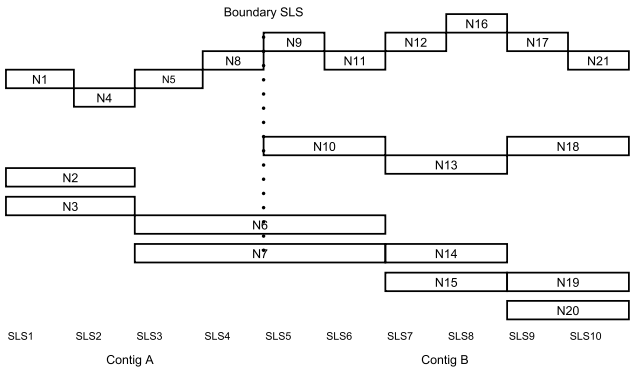


Figure 5. Contig Segmentation within a piece after vertical integration.

Thus, we propose the use of the contig mapping approach according to the cluster count as an additional step between the vertical and horizontal integration processes. Formally, if C_t represents the cluster count at SLS_t, the boundary between time slices $t-1$ and t becomes a segmentation boundary if:

1. $C_t \neq C_{t-1}$, or
2. $C_t = C_{t-1}$, in which case the cluster status changes.

The status change is caused by overlapping clusters that cross over an SLS that has been marked as a segmentation boundary. In this case, the overlapping clusters are separated at SLS_t into two clusters with the same pitch and overall duration as the initial. Figure 5 shows an example of contig segmentation. Until SLS_4 the cluster count is 2 within $Contig_A$. At SLS_5 the cluster count has not changed but an overlap cluster from previous SLS does exist. The cluster with notes {N6, N7} will be thus separated into two clusters. Thus, {N6a, N7a} will have onset as in SLS_3 and offset as in SLS_5, while {N6b, N7b} will have onset as in SLS_5 and offset as in SLS_7, respectively.

3.3 Stream Matching

As mentioned in Section 2.1.1, after determining the clusters for each SLS, a bipartite graph is created for matching notes to streams. Each cell (i, j) of the graph designates the cost between the last cluster assigned to stream i and the current cluster j. The previous versions of the VISA algorithm moved in a step-wise fashion through the input sequence, creating the graph and then assigning the streams. The following factors were used for the calculation of the cost:

1. Homogeneity factor 25%: Refers to the difference of the number of notes between clusters. Consecu-

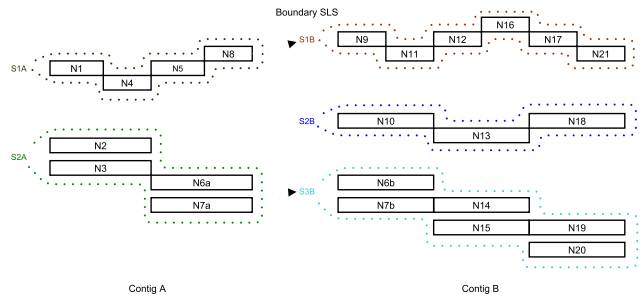


Figure 6. Stream Matching between consecutive contigs.

tive clusters with the same number of notes are more likely to belong to the same stream.

2. Pitch Proximity 50%: Distinguishes the clusters that have close average pitch with the available streams.
3. Temporal Continuity 25%: Music rests (gaps) between consecutive clusters impose additional cost for the assignment.

In $VISA^3$, we propose the same factors but with slightly different methodology:

1. Assign streams in every contig: The number of clusters C_t in a contig represents the number of streams V_t.
2. Integrate the contigs by calculating the assignment costs on all segmentation boundaries: If at SLS_t holds that $C_t \neq C_{t-1}$, then this is the end of contig Cg_{t-1} and the beginning of Cg_t. In order to connect the streams we calculate the cost using the same factors, as mentioned before, between the last clusters assigned to stream $i \in V_{t-1}$ of Cg_{t-1} and current clusters of Cg_t.

Figure 6 presents a scenario of stream assignment between contigs based on the previous example. $Contig_A$ has cluster count 2, and therefore 2 streams, $S1_a$ and $S1_b$, were assigned to all its clusters. Similarly, $Contig_B$ has 3 streams. The connection between the streams $S1_x$ and $S2_x$ is based on a stream assignment of the first clusters of $Contig_B$ with the streams of the last assigned clusters of $Contig_A$. For example, the cluster consisting of the note $N9$ is more likely to connect with a stream in which the last cluster assigned is $N8$. Therefore, a link exists between $S1_A$ and $S1_B$. Finally, it is worth mentioning that the homophonic procedure that forces the algorithm to switch to the two basic streams, as described in previous versions of the VISA algorithm, is completely removed in $VISA^3$ as it is not required due to the use of the Contig Clustering process.

4. PERFORMANCE EVALUATION

This section presents a concise description of the experimentation platform and data sets, followed by a performance analysis based on experimentation on the proposed method. The implementation is under MATLAB with the use of MIDIToolbox [17] for auxiliary functions.

4.1 Experimental Set-up

The proposed algorithm has been tested with two different datasets of quantised symbolic data. The first dataset consists mostly of the same data with the VISA09 version, for the purposes of comparing/contrasting the performance of VISA09 and $VISA^3$. It includes 30 pieces, featuring 16 excerpts primarily from piano sonatas by Beethoven, seven fugues and inventions by J.S.Bach, three mazurkas and two waltzes by F.Chopin. The selection of these pieces was intended to capture diverse musical textures, i.e. homophonic and contrapuntal textures. The majority of these pieces contain homophonic texture with two streams, consisting of a melody (upper staff) and accompanying harmony (lower staff). J.S. Bach's pieces feature independent monophonic streams, while very few pieces from Beethoven include parallel movement cases.

In order to further expand the testing corpus, we created a second small dataset with a selection of traditional Greek folk popular music. 30 MIDI files from the *Greek Music Dataset*, a freely available collection of features and metadata for 1400 popular Greek tracks [18], were selected randomly to expand the experimental examination corpus. After pre-processing, which included the deletion of duplicate instrument tracks and drum tracks, only pieces with different polyphonic and monophonic independent streams were kept. Then, an annotation task was conducted by a music theory research student that was aimed to identify streams in the scores after listening each excerpt. A number of musical examples which contained parallel movement cases, homophonic and polyphonic textures were discussed with the expert before doing this task. Therefore, bearing in mind all the above restrictions, the total number of the annotated tracks was reduced to 14.

The evaluation metric used herein is the precision of the obtained result. Herein, precision refers to the sum of notes that have been correctly assigned to the appropriate stream (according to the ground-truth), divided by the total number of notes.

4.2 Results

Table 1 shows the complete results of the proposed methodology for both datasets. The average precision of VISA09 in the classical dataset is 82,1% while with the proposed refinement, $VISA^3$ reaches 88,9%. An even more notable amelioration in precision is detected in the popular dataset where VISA09's precision is 62,8% while $VISA^3$ achieves 80,5%. Accordingly, the proposed modifications into the VISA family offer significant improvement as far as the performance of the algorithm is concerned.

More specifically, $VISA^3$ improves the precision on pieces where non-parallel movement is detected according to the *Co-Modulation Principle*, in both datasets. Accordingly, we present two examples by providing the score and the corresponding pianorolls as well as with the ground truth, for both VISA09 and $VISA^3$ assignment. Each color on the pianoroll corresponds to different stream. Figure 7 presents one such example wherein VISA09 detects two streams on the first bar, considering only the *Synchronous Note Principle*. On the other hand, $VISA^3$ detects three

	VISA09	$VISA^3$
Classical Dataset		
Beethoven, Sonata 2-1 Prestissimo	93.0%	93.6%
Beethoven, Sonata 2-1 Adagio	83.0%	86.8%
Beethoven, Sonata 2-2 AllegroVivace	79.8%	85.1%
Beethoven, Sonata 2-2 LargoApp	91.0%	95.3%
Beethoven, Sonata 2-2 Rondo	82.0%	83.9%
Beethoven, Sonata 2-2 Scherzo	75.0%	95.3%
Beethoven, Sonata 2-3 Adagio	77.0%	89.1%
Beethoven, Sonata 2-3 AllegroAssai	94.0%	98.6%
Beethoven, Sonata 2-3 AllegroConBrio	87.0%	87.3%
Beethoven, Sonata 2-3 Scherzo	73.0%	75.9%
Beethoven, Sonata 10-2 Allegretto	73.0%	90.1%
Beethoven, Sonata 10-2 Allegro	89.0%	97.2%
Beethoven, Sonata 10-2 FinalePresto	92.0%	100%
Beethoven, Sonata 13 AdagioCantabile	47.7%	78.0%
Beethoven, Sonata 13 Grave	97.9%	93.4%
Beethoven, Sonata 13 Rondo	85.0%	87.7%
Brahms, Waltz Op39 No8	89.0%	96.5%
Bach, Fugue BWV 852	91.0%	89.7%
Bach, Fugue BWV 856	94.0%	85.4%
Bach, Fugue BWV 772	96.7%	97.4%
Bach, Fugue BWV 784	93.4%	95.0%
Bach, Fugue BWV 846	49.6%	77.4%
Bach, Fugue BWV 859	32.8%	78.2%
Bach, Fugue BWV 281	39.2%	56.5%
Joplin, Harmony Club Waltz	92.3%	89.5%
Chopin, Waltz Op64 No1	91.2%	91.0%
Chopin, Waltz Op69 No2	96.2%	92.1%
Chopin, Mazurka Op7 No1	92.4%	90.8%
Chopin, Mazurka Op7 No5	96.6%	100%
Chopin, Mazurka Op67 No4	89.6%	91.3%
Popular Dataset (ID Tags)		
Marinella - Agaph pou egines dikopo maxairi ID 267	58.1%	74.4%
Marinella - Stalia, Stalia ID 10	38.1%	87.1%
Grhgorhs Bithikwtshs - Asprh Mera kai gia emas ID 385	85.6%	95.3%
Markos Vamvakarhs - Mikros Aravwniastika ID 1004	73.7%	95.1%
Mikis Theodwrakhs - Tis dikaiosynhs hlie nohte ID 1053	70.6%	81.0%
Maria Dhmhtriadh - To treno feugei stis 8 ID 1057	77.7%	88.7%
Kaith Xwmata - Ki an se agapw den se orizw ID 1240	77.6%	87.8%
Dhmhtra Galanh - Vre pws allazoun oi kairoi ID 1295	24.2%	59.9%
Vasilhs Tsitsanhs - Gia ta matia pou agapw ID 1256	65.9%	65.0%
Vasilhs Tsitsanhs - Mpakse tsifliki ID 1274	60.6%	74.7%
Vasilhs Tsitsanhs - Trekse magka na rwthseis ID 1290	32.7%	77.6%
Alikh Vougiouklakh - Gaidarakos ID 1320	71.3%	77.1%
Grhgorhs Bithikwtshs - Eimai aetos xwris ftera ID 1322	80.9%	87.2%
Mairh Lw - Epta tragoudia tha sou pw ID 1325	62.5%	75.5%

Table 1. Precision for stream separation by the previous and the current implementation of VISA on the Classical and Popular Dataset.

streams, since the top and bottom notes move in non-parallel fashion. In the second bar, both versions find 3 streams due to different note durations in every SLS while in the third bar, similarly to the case of the first bar, VISA09 detects only two of the three streams by considering solely the *Synchronous Note Principle*.

Another representative example with non-parallel movement is shown in Figure 8 where the texture can be characterised as homophonic. VISA09, when detecting homophonic texture, forces the use of one stream, i.e. all synchronized notes are assigned to one chordal stream (or two streams, i.e. main melody notes and accompaniment if melody contains some different note durations). VISA09 does not check for parallel movement in homophonic clusters and, therefore, does not have the ability to identify streams due to different motion within homophony. In this instance, it fails to recognize the three streams indicated in the ground truth, and therefore the precision is very low.

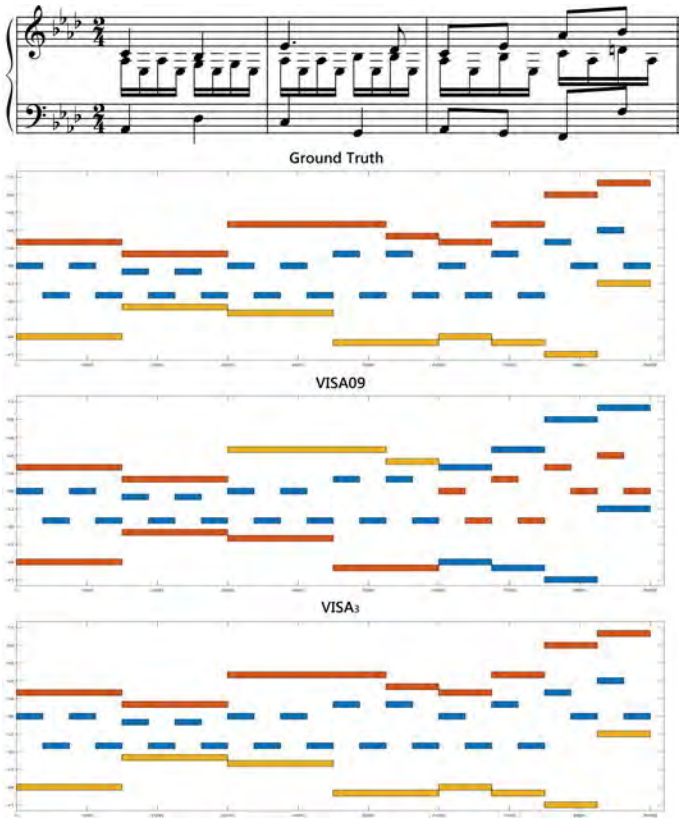


Figure 7. Opening of Beethoven's, Sonata 13, Adagio Cantabile.

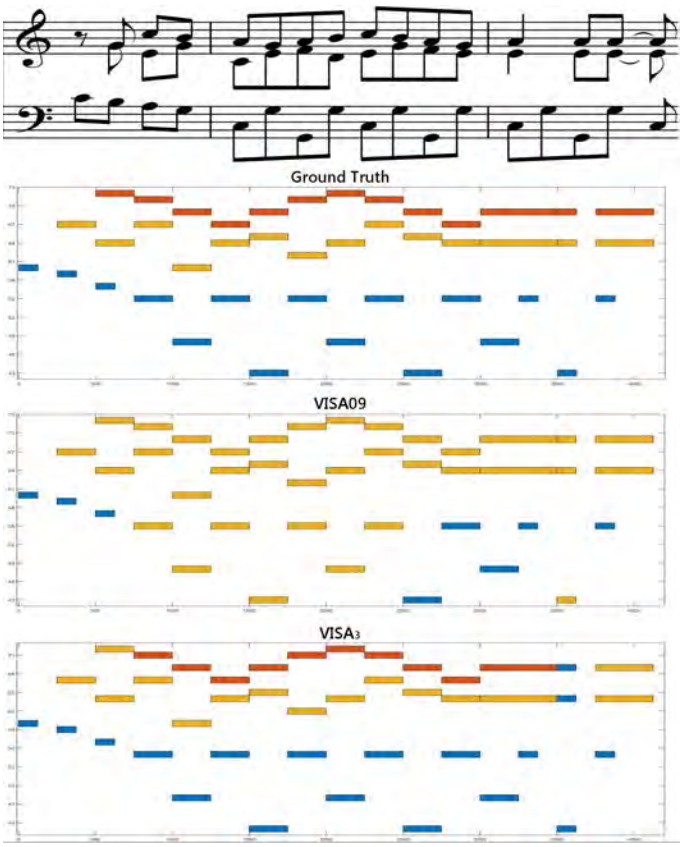


Figure 8. Opening of the Greek folk song *Alikh Vougiouklakh - Gaidarakos*.

In contrast, $VISA^3$ achieves far better results by detecting correctly the non-parallel movement between consecutive clusters and separates these to different streams. Furthermore, considering the contig segmentation of the piece, the algorithm is not carrying further initial wrong stream assignments. As shown on the assignment results for $VISA^3$ in Figure 8, $VISA^3$ fails to separate the single clusters containing two or three notes, though as the cluster count changes, a new contig begins and the stream assignment continues smoothly without taking into account previous errors.

5. CONCLUSIONS

This work presents the $VISA^3$ algorithm, a refinement of the family of VISA algorithms for integration/segregation of voice/streams. $VISA^3$ builds upon its previous editions by discarding unnecessary techniques and introducing new that adhere to general perceptual principles, address accumulation errors and tackle more generic musical content. Moreover, a new small dataset of quantised symbolic data with human-expert ground-truth annotation is utilised.

Experimental results indicated that the proposed algorithm achieves significantly better performance than its predecessors. The increase in precision is evident for both the dataset of the previous editions as well as for a new dataset that includes musical content with characteristics such that of non-parallel motion that are common and thus required to be addressed.

Future plans include the examination of alternative methods to avoid early stream assignment error propagation, less strict evaluation measurements such as customisations of the Note-based [8] and Transition-based [19] evaluation metrics used in voice separation tasks as well as and the expansion of the ground-truth dataset with more diverse musical content.

6. REFERENCES

[1] I. Karydis, A. Nanopoulos, A. Papadopoulos, E. Cambouropoulos, and Y. Manolopoulos, "Horizontal and vertical integration/segregation in auditory streaming: A voice separation algorithm for symbolic musical data," in *Sound and Music Computing Conference*, 2007, pp. 299–306.

[2] E. Cambouropoulos, "Voice and stream: Perceptual and computational modeling of voice separation," *Music Perception*, vol. 26, no. 1, pp. 75–94, 2008.

[3] D. Rafailidis, E. Cambouropoulos, and Y. Manolopoulos, "Musical voice integration/segregation: VISA revisited," in *Sound and Music Computing Conference*, 2009, pp. 42–47.

[4] E. Cambouropoulos, "'Voice' separation: theoretical, perceptual and computational perspectives," in *International Conference on Music Perception and Cognition*, 2006, pp. 987–997.

[5] N. Guiomard-Kagan, M. Giraud, R. Groult, and F. Leve, "Comparing voice and stream segmentation algorithms," in *International Society for Music Information Retrieval Conference*, 2015, pp. 493–499.

[6] D. Temperley, *The Cognition of Basic Musical Structures*. MIT Press, 2001.

[7] E. Cambouropoulos, "From midi to traditional musical notation," in *AAAI Workshop on Artificial Intelligence and Music: Towards Formal Models of Composition, Performance and Analysis*, 2000.

[8] E. Chew and X. Wu, *Computer Music Modeling and Retrieval: Second International Symposium, CMMR 2004, Esbjerg, Denmark, May 26-29, 2004. Revised Papers*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2005, ch. Separating Voices in Polyphonic Music: A Contig Mapping Approach, pp. 1–20.

[9] W. M. Szeto and M. H. Wong, "A stream segregation algorithm for polyphonic music databases," in *Database Engineering and Applications Symposium*, 2003, pp. 130–138.

[10] J. Kilian and H. H. Hoos, "Voice separation - a local optimisation approach," in *International Conference on Music Information Retrieval*, 2002, pp. 39–46.

[11] I. Karydis, A. Nanopoulos, A. N. Papadopoulos, and E. Cambouropoulos, "VISA: The voice integration/segregation algorithm," in *International Society for Music Information Retrieval Conference*, 2007, pp. 445–448.

[12] A. S. Bregman, *Auditory scene analysis: The perceptual organization of sound*. MIT press, 1990.

[13] E. Narmour, *The analysis and cognition of melodic complexity: The implication-realization model*. University of Chicago Press, 1992.

[14] D. Huron, "Tone and voice: A derivation of the rules of voice-leading from perceptual principles," *Music Perception: An Interdisciplinary Journal*, vol. 19, no. 1, pp. 1–64, 2001.

[15] D. Rafailidis, A. Nanopoulos, Y. Manolopoulos, and E. Cambouropoulos, "Detection of stream segments in symbolic musical data," in *International Society for Music Information Retrieval Conference*, 2008, pp. 83–88.

[16] R. D. Morris, "Voice-leading spaces," *Music Theory Spectrum*, vol. 20, no. 2, pp. 175–208, 1998.

[17] T. Eerola and P. Toiviainen, "Mir in matlab: The midi toolbox," in *International Society for Music Information Retrieval Conference*, 2004.

[18] D. Makris, I. Karydis, and S. Sioutas, "The greek music dataset," in *Proceedings of the 16th International Conference on Engineering Applications of Neural Networks (INNS)*. ACM, 2015, p. 22.

[19] P. B. Kirlin and P. E. Utgoff, "Voise: Learning to segregate voices in explicit and implicit polyphony." in *ISMIR*. Citeseer, 2005, pp. 552–557.

ADAPTING A COMPUTATIONAL MULTI AGENT MODEL FOR HUMPBACK WHALE SONG RESEARCH FOR USE AS A TOOL FOR ALGORITHMIC COMPOSITION

Michael Mcloughlin
Plymouth University,
Interdisciplinary Centre for Computer Music Research, Plymouth, UK
michael.mcloughlin@-Plymouth.ac.uk

Luca Lamoni
University of St. Andrews
School of Biology, St. Andrews, UK
ll42@st-andrews.ac.uk

Ellen Garland
University of St. Andrews
School of Biology, St. Andrews, UK
ecg5@st-andrews.ac.uk

Simon Ingram
Plymouth University,
School of Marine Science and Engineering, Plymouth, UK
simon.ingram@-Plymouth.ac.uk

Alexis Kirke
Plymouth University,
Interdisciplinary Centre for Computer Music Research, Plymouth, UK,
Alexis.kirke@plymouth.ac.uk

Michael Noad
University of Queensland,
Cetacean Ecology and acoustics Laboratory, School of Veterinary Science, Gatton, Australia
mnoad@uq.edu.au

Luke Rendell
University of St. Andrews
School of Biology, St. Andrews, UK,
ler4@st-andrews.ac.uk

Eduardo Miranda
Plymouth University,
Interdisciplinary Centre for Computer Music Research, Plymouth, UK,
Eduardo.Miranda@-Plymouth.ac.uk

ABSTRACT

Humpback whales (*Megaptera Novaengliae)* present one of the most complex displays of cultural transmission amongst non-humans. During breeding seasons, male humpback whales create long, hierarchical songs, which are shared amongst a population. Every male in the population conforms to the same song in a population. During the breeding season these songs slowly change and the song at the end of the breeding season is significantly different from the song heard at the start of the breeding season. The song of a population can also be replaced, if a new song from a different population is introduced. This is known as song revolution. Our research focuses on building computational multi agent models, which seek to recreate these phenomena observed in the wild. Our research relies on methods inspired by computational multi agent models for the evolution of music. This interdisciplinary approach has allowed us to adapt our model so that it may be used not only as a scientific tool, but also a creative tool for algorithmic composition. This paper discusses the model in detail, and then demonstrates how it may be adapted for use as an algorithmic composition tool.

1. INTRODUCTION

Multi agent modelling is a powerful tool where autonomous artificial intelligences (agents), interact with each other and their environment. As they interact, they can produce emergent behaviour, and create phenomena that are not built directly into the system. This has made it a powerful tool in scientific research where it has been used to study the emergence of grammar in linguistics [1], genetic diversity in humans [2], and flocking behaviour in birds and fish [3].

Due to the emergent phenomena produced by multi agent models, they have found use in several different areas of sound and music computing. From a musicology perspective, research shows that they may be used to explain a variety of phenomena, from songs emerging from sexual selection pressure [4] to the evolution of intonation systems [5]. Aside from musicological research, multi agent modelling is used as a tool for algorithmic composition [6]. In this paper, we seek to demonstrate that the gap between multi agent modelling for scientific purposes and for creative purposes is often quite narrow.

The model presented here was originally designed to investigate the mechanisms underlying cultural transmission in humpback whales. We show it is possible to adapt this model and use it as a tool for algorithmic composition. First, an overview of the structure of humpback whale song is introduced, followed by a description of our model and its aims. Then, an in depth analysis of the model is presented, outside of the context of algorithmic composition. Finally, we describe the method used to adapt the model as a tool for composition and give an example of user interaction with the model.

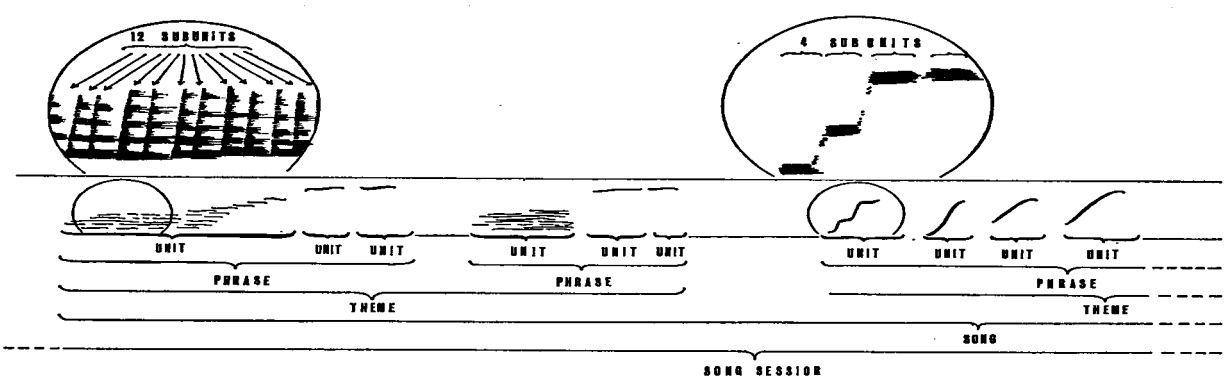


Figure 1: The structure of humpback whale song. Taken from [7]. Time on the X-axis and frequency on Y-axis.

2. HUMPBACK WHALE SONG

Before investigating the model in detail, it is necessary to first describe the natural phenomena that the model seeks to recreate. Humpback whales are a species of baleen (Mysticeti) whale. During the summer months (feeding season), humpback whales are usually found in polar Regions where the plankton that the whales feed on is abundant. During the winter months (breeding season), they migrate to warmer, tropical waters. Here they mate and give birth to calves. During migration and on the breeding grounds, male humpbacks produce long, hierarchical vocal sequences termed 'songs' [7]. Males produce individual sounds called units, which are combined to create phrases. Phrases are then combined to create themes and themes are stringed together to create songs. Songs are then repeated to create song sessions. During the mating season, all males conform to the same song. The song gradually changes throughout the season. This slow change is known as 'song evolution' [8]. It is also possible for the song of a population to be replaced by the song of another population. This is known as song revolution [9]. This was first observed when the song of the western Australian population replaced the song of the eastern Australian population. Further research into the population east of Australia revealed that this revolutionary behaviour was not an isolated incident, as the song of the eastern Australian population took over the song of the New Caledonia population. This song continued to move eastward until it eventually took over the song of the French Polynesian population [10].

Understanding these phenomena is vital, as it is what we seek to recreate using our model. Specifically, our goals are to create a spatially explicit multi agent models, where populations evolve shared repertoires, but also present the possibility for new songs to be introduced and replace the existing songs of a population.

3. THE MODEL

The model is cyclic in nature, and is segmented into three sequential sections; movement rules, song production rules, and song learning rules. These rules describe the behavior of a single agent and are carried out for every agent. Our model is created in Python using the SciPy package [11]. This model was inspired by [12] and extends on research done in [13].

3.1 Movement Rules

When our model is initialized, agents are assigned random Cartesian co-ordinates within a certain area. This area is known as the feeding grounds. While on the feeding grounds they carry out random walks to navigate the plane. After a certain number of iterations specified by the user, the agents will migrate to the 'breeding grounds', the location of which are also specified by the user. The movement behaviour to and on the feeding grounds is controlled by a variety of rules inspired by flocking algorithms. To explain these rules, we examine them from the perspective of a single agent. Our focal agent has two areas around it, a Zone of Repulsion (ZOR) and a Zone of Attraction (ZOA), as shown in Figure 2.

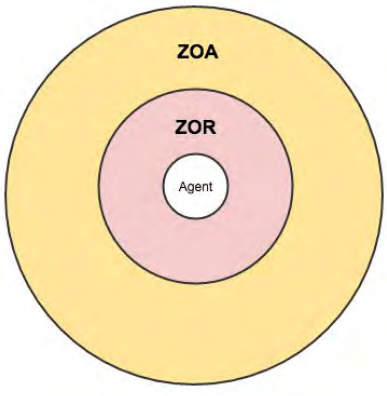


Figure 2: The two different zones around an agent. When other agents enter these zones, certain movement rules are carried out.

The ZOR rule is enacted whenever an agent has other agents within its ZOR. When this happens, an agent calculates a new trajectory based on the position of the other agents within its ZOR. This is demonstrated in Figure 3. This rule is carried out each iteration of the model.

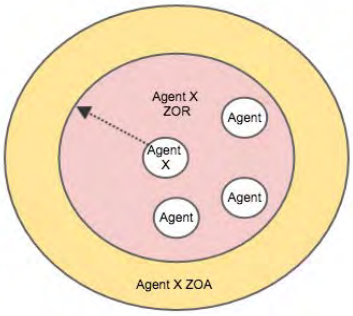


Figure 3: The ZOR rule. Agent x has other agents within its Zone of Repulsion. It calculates a new trajectory in order to avoid these agents.

The ZOA rule is enacted only when an agent is within a certain distance of the breeding grounds. This rule causes an agent to approach whatever agent within its ZOA that has the song most similar to its own. This is achieved using Levenshtein Distance. This algorithm calculates the number of insertions, substitutions and deletions required to transform one string of symbols into another string of symbols. Using these values, we calculate a ratio of similarity between two sequences of symbols produced by our agents. This rule is inspired by interactions between male humpback whales on the breeding ground [14].

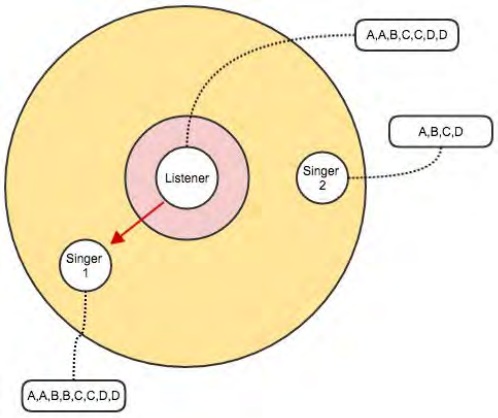


Figure 4: The attraction rule. The listening agent moves towards the singer with most similar song to its own.

3.2 Song Production Rules

Agents in the model are equipped with a first order transition matrix that is used to generate new songs. In our model, songs are represented using integers. Each integer corresponds with a unit that is associated with humpback whale song. Songs are generated from this transition matrix using equation 1.

$$x = \sum c \leq U \tag{1}$$

Where x is the output unit, c is the cumulative summation of the probability vector (the row of our transition matrix we are currently sampling from), and U is a uniformly distributed random number between 0 and 1. We use this algorithm in a recursive function to generate songs of varying length.

3.3 Song Learning Rules

In our initial model, song learning is affected only by the distance between agents. At every model run, an agent will calculate its distance from every other agent in population using the Cartesian distance formula, in equation 2.

$$d = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2} \tag{2}$$

Where x_1 and y_1 are our focal agents Cartesian coordinates and x_2 and y_2 are the co-ordinates of the agent we wish to calculate the distance for, and d is distance. We use d to calculate what we call the intensity factor, which represents the energy decay in the water. It is calculated in equation 3.

$$I = \frac{1}{d^2} \tag{3}$$

Where I represents the intensity factor, and d is the distance between the two agents we are calculating I for. We can now go about the song learning stage. First, an agent estimates a transition matrix for an input sequence. This input sequence is simply the song produced by another agent in our population. To update the listening agents new transition matrix, we carry out the following matrix weighted averaging function in equation 4.

$$T_A = \big(A * (1 - I)\big) + (B * I) \tag{4}$$

Where T_A is the updated transition matrix for the listening agent, A is the original transition matrix for the listening agent, B is the estimated transition matrix for the sequence produced by a singing agent, and I is the intensity factor.

4. MODEL RESULTS

For a quick qualitative analysis, we plot our agent's Cartesian tracks and the distance between the songs of each agent using a Levenshtein distance dendrogram. This is shown in Figure 5 and Figure 6. This shows two possible scenarios that may emerge after running the model. In figure 5, we can see that the agents have clustered and have begun moving together throughout the breeding grounds. Due to the distance bias, every agent has converged on the same song. Figure 6 presents a similar situation, except the agents have formed into three distinct clusters, with three different songs emerging. This echoes results observed in the wild, where distinct populations converge on distinct songs.

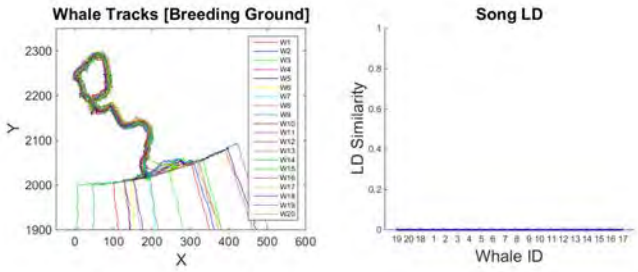


Figure 5: An example of all agents moving together and converging on the same song. The diagram on the left are the Cartesian co-ordinates of the agents over the run of the model. The diagram on the left is a dendrogram showing the level of dissimilarity between the songs of every agent.

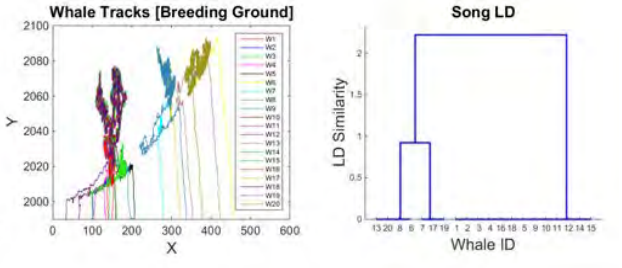


Figure 6: An example of the agents forming separate groups, each with their own song, as can be seen by the dendrogram.

While this result is interesting, it is necessary to understand how varying the parameters in the model will affect the songs produced by the population. To achieve this, we created a series of 500 experiments, in which we used linear descent to vary the size of the ZOR, ZOA, breeding grounds, and feeding grounds. We then carry out a pairwise subtraction of every agent's transition matrix from each other. This allows us to see whether the agents have converged on the same transition matrix or if there is a large amount of variety in them. These results were then stored in a 100x5 matrix. Each column corresponded with the size of the ZOA, and every 25 rows corresponded with an increase in the ZOR. Within those 25 rows and column is every combination of feeding and breeding ground size. We then took the mode of every group of these cells. This resulted in the matrix seen in Figure 7.

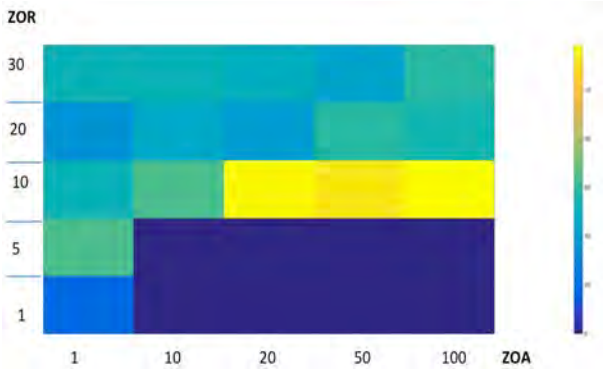


Figure 7: This graph shows how varying the parameters of the ZOR and the ZOA affects the behaviour of the model. The darker colours correspond to model runs that converged on similar songs. The bright colours represent experiments where the population had dis-similar songs.

5. ADAPTING THE MODEL

As it stands, the model does not capture the full complexity observed in humpback whale song. While agents in our model do converge on a shared song in certain situations, it does not present any change once every agent in the population has learned the song. Furthermore, first order transition matrices are not capable of capturing the hierarchical structure of humpback whale song. Despite this, our model is at a stage where it can be adapted for algorithmic composition. In this section, we describe the technical aspects of adapting our model. Following this, we move on to discuss adding a novelty method inspired by computational musicology.

5.1 Technical Considerations

Open Sound Control (OSC) [15] is a protocol used to transmit data between different audio software programs. This allows for the quick adaptation of our model to be used as a tool for composition. Using OSC, we can send data to and from our model in order to generate new musical sequences in real time. This is achieved using the Max4Live API in Ableton Live[16,17], so that the composer may introduce new songs sequences to the population, and play them back in order to generate new musical variations, based on this input and other parameter settings. This is illustrated in Figure 8.

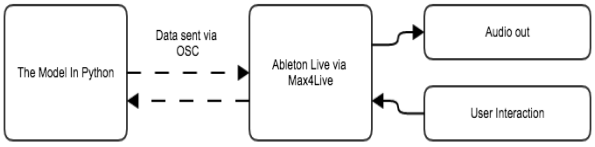


Figure 8: Signal flow from Ableton to the model.

In order to interact with the model, the user uses the ComposerIn device with a MIDI keyboard to create a sequence of notes and rhythms to be learned by a selected agent in the model. These notes are appended to a list in Max/MSP, where they are then formatted so that they can be used as an input to the model. The sequence is then sent via OSC to the selected agent, who estimates a first order transition matrix so that it may create variations on this theme. This interaction flow is demonstrated in figure 9.

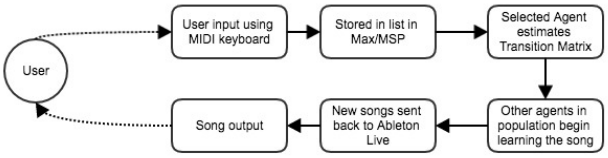


Figure 9: This shows how a composer interacts with the model.

At each iteration of our model, the song produced by each agent is sent back to Ableton Live using the modelOut device, where they are transposed in order for them to be formatted into MIDI notes. These are then stored in a message box and sequenced using a metro object. This allows the MIDI notes to be sent to any Live or Max4Live device that the composer wishes to use.

5.2 The Need for Novelty

Since the model relies on the distance between agents to influence the transmission of the song, it is possible for the song input by the composer to be overpowered by the other songs in the population of agents. This leads us to add a new dimension to the model, which will allow the user to interact with the model and actually observe the impact of their input. To achieve this, the model is extended to have a new bias added; novelty. Chosen due to theories that is a factor in humpback song evolution. [9]

Originally, we took inspiration from the work of Todd, [18] where novelty is determined by the built in expectations of the agents. This was used to develop in equation 5.

$$\alpha = \sum_{n}^{N} \frac{\left|\max\left(T(S(n))\right) - T(S(n), S(n+1))\right|}{N} \quad (5)$$

Given a sequence, *S,* which is indexed using the value *n,* an agent calculates novelty, α, based on its transition matrix, *T. N,* the number of elements in the sequence *S,* is used as a weighting. This is defined by equation 5. The square brackets indicate that an absolute value should be taken for the top term of the equation. This novelty value, α, is used to update our learning algorithm, as shown in equation 6.

$$T_A = (A * (1 - I * \alpha)) + (B * (I * \alpha)) \quad (6)$$

The novelty bias has a significant impact on what songs our agents choose to learn from. Low novelty will result in a song having no impact on the transition matrix of a listening agent, while a high novelty value will lead to that songs estimated transition matrix almost completely taking over the listening agent's transition matrix (Figure 10).

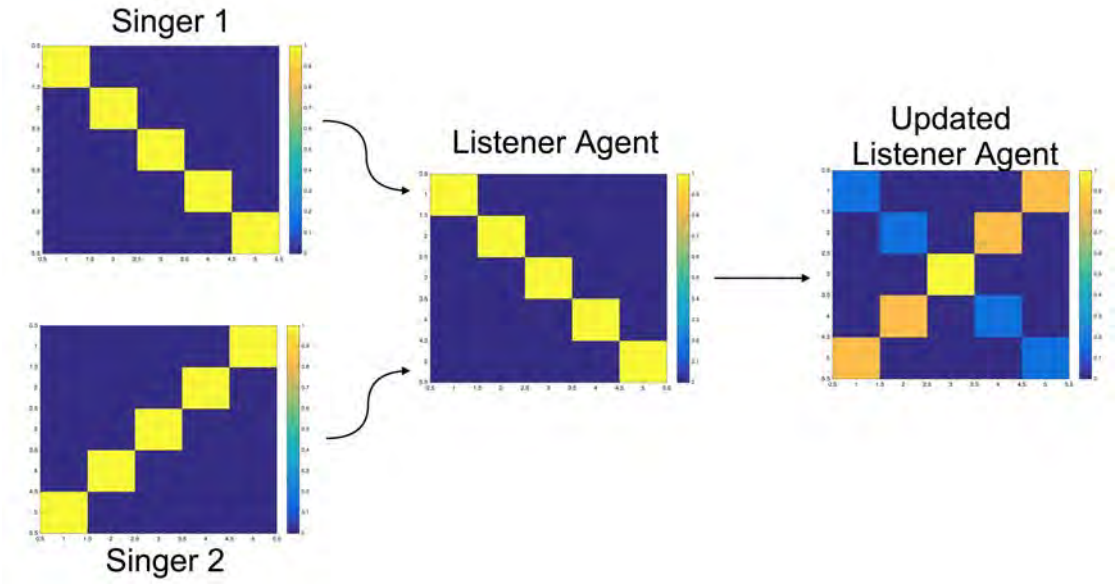


Figure 10: How the novelty algorithm affects a listener agent's transition matrix. Singer 1 and 2 are both equidistant from the listener agent. The listener learns more of singers 2 song since it is more novel in comparison to its own transition matrix.

The results returned from this method are interesting, but we found that the transition matrices in a population would converge to uniform distribution. Rather than our novelty value being weighted by the number of elements in a sequence, we have our novelty value weighted by the agent that has the highest novelty value.

$$\text{nov[m]} = \sum_n^N |\max\left(T(S(n)) - T(S(n), S(n+1))\right| \quad (7)$$

$$\alpha = \frac{\text{nov(m)}}{\max(nov)} \quad (8)$$

We also introduce a learning rate with this algorithm, which is scaled between 0 and 1. This updates our agents learning algorithm to the following, seen in equation 13.

$$T_A = (A * (1 - I * (\alpha * \text{LR})) + (B * (I * (\alpha * \text{LR})) \quad (9)$$

The dynamic weighting algorithm produces an oscillating effect on the probability of transitioning from one unit to another, as demonstrated in Figure 11. This shows the probability of an agent moving from unit A to unit B (the red line), and the probability of moving from unit A to unit C (the blue line). At the start of the model, the probability of going from unit A to B is 100%. Another agent in the population is trained with a probability of transferring from unit A to C 100% of the time. As our agents meet on the breeding grounds they hear the song, they hear this new song and deem it to be more novel than their own, thus applying more emphasis to learning it.

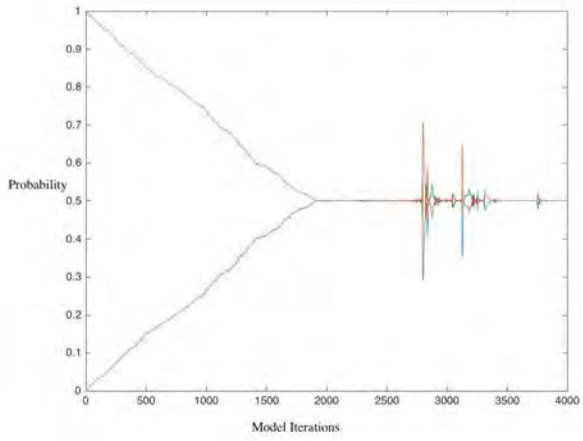


Figure 11: The figure demonstrates how the dynamic novelty algorithm creates an oscillation in the probability of transitioning from one unit to the other. (Transitioning from unit 1 to 2 in blue, transitioning from unit 1 to 3 in red).

6. MUSICAL DEMO

In order to test the model, we approached it from a compositional point of view. First, four different musical themes were chosen to form the structure of the composition. These themes were chosen specifically because they have a high novelty value when compared to each other. They are also easily recognisable rudimentary musical features. They consist of an ascending C major arpeggio (Theme A), a descending chromatic scale (Theme B), an ascending D minor arpeggio (Theme C), and a repeating G# (Theme D). These themes can be seen in figure 12. At the start of our composition, every agent's transition matrix is trained by theme A. We then presented all subsequent themes to only a single agent (agent 2). We then recorded the songs being produced by agent 1 via MIDI.

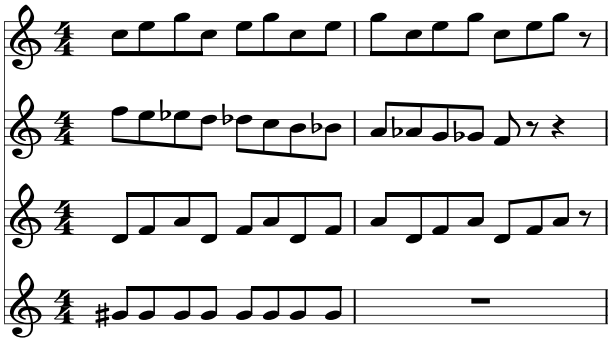


Figure 12: The four themes used in the composition.

The resulting composition is interesting, as the oscillatory nature described in section 5.2 of this document emerged not only for simple transitions as was originally observed, but also for the structured themes presented to our population. Whenever a new theme was introduced, the agent would move between the two themes, before the entire population would settle on some form of hybrid theme. We called this theme oscillation (figure 14). The corresponding hybrid theme is shown in figure 13. This is demonstrated at the point where each transition is introduced.

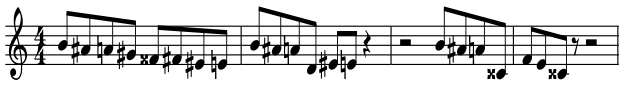


Figure 13: An example of a hybrid theme.

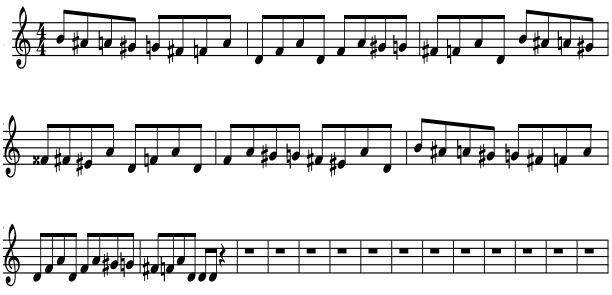


Figure 14: An example of theme oscillation.

7. CONCLUSION AND FUTURE WORK

From this paper, we have seen that scientific methods for the analysis of animal vocalisations may easily be adapted for algorithmic composition. Here, we demonstrated the model as a stand alone scientific tool, ex-

plained the technical considerations necessary for user interaction, and developed a novelty method that allows a user to have a direct impact on the songs in the population. Emergent properties, such as theme oscillation and hybrid themes were also demonstrated through a compositional demo. Future work will involve exploring the parameter space described in section 4, to investigate how it may be used as a tool to influence the emergence of hybrid themes seen in this model. It is also necessary to carry out a full investigation of the impact that the novelty metric has on the evolution of songs in the population. Finally, a method of song innovation must be produced. Although our agents develop interesting hybrid songs, they do not have any in built mechanism for song evolution. The use of the model to develop rhythmic themes is also an area that would warrant further investigation.

Acknowledgments

The authors wish to thank The Leverhulme Trust for funding this project.

8. REFERENCES

[1] S. Kirby, "Spontaneous evolution of linguistic structure - An iterated learning model of the emergence of regularity and irregularity," *IEEE Trans. Evol. Comput.*, vol. 5, no. 2, pp. 102–110, 2001.

[2] H. Whitehead, P. J. Richerson, and R. Boyd, "Cultural Selection and Genetic Diversity in Humans," *Selection*, vol. 3, pp. 115–125, 2002.

[3] C. Hartman and B. Beneš, "Autonomous boids," in *Computer Animation and Virtual Worlds*, 2006, vol. 17, no. 3–4, pp. 199–206.

[4] E. R. Miranda, S. Kirby, and P. Todd, "On Computational Models of the Evolution of Music: From the Origins of Musical Taste to the Emergence of Grammars," *Contemporary Music Review*, vol. 22, no. 3. pp. 91–111, 2003.

[5] E. R. Martins, Joao M., Miranda, "Engineering - The Role of Social Pressure: A New Artificial Life Approach to Software for Generative Music," *i-managers J. Softw. Eng.*, vol. 2, no. 3, 2008.

[6] E. R. Miranda, Ed., *A-Life for Music: Music and Computer Models of Living Systems*. Middleton, Wisconson: A-R Editions, Inc., 2011.

[7] R. S. Payne and S. McVay, "Songs of humpback whales.," *Science*, vol. 173, no. 3997, pp. 585–597, 1971.

[8] R. Payne, K., Tyack, P., & Payne, "Progressive changes in the songs of humpback whales (Megaptera novaeangliae): A detailed analysis of two seasons in hawaii," in *Communication and Behavior of Whales*, no. 9–57, 1983, pp. 9–57.

[9] C. Noad, Michael J., Cato, Douglas H., Bryden, M. M., Micheline, Jenner, Jenner, "Cultural Revolution in Whale Songs," *Nature*, vol. 408, no. 537, 2000.

[10] E. C. Garland, A. W. Goldizen, M. L. Rekdahl, R. Constantine, C. Garrigue, N. D. Hauser, M. M. Poole, J. Robbins, and M. J. Noad, "Dynamic horizontal cultural transmission of humpback whale song at the ocean basin scale," *Curr. Biol.*, vol. 21, no. 8, pp. 687–691, 2011.

[11] P. Jones, Eric and Oliphant, T. and Peterson, "SciPy: Open Source Scientific Tools for Python." 2001.

[12] A. Kirke, S. Freeman, E. Miranda, and S. Ingram, "Application of Multi-Agent Whale Modelling to an Interactive Saxophone and Whales Duet," in *Proceedings of International Computer Music Conference*, 2011, no. August, pp. 350–353.

[13] A. Kirke, E. Miranda, L. Rendell, and S. Ingram, "Towards Modelling Song Propagation in Humpback Whales," in *Proceedings of 2015 Conference on Transdisciplinary Approaches to Cognitive Innovation*, 2015.

[14] J. D. Darling, M. E. Jones, and C. P. Nicklin, "Humpback whale songs : Do they organize males during the breeding season ?," *Behaviour*, vol. 9, no. 143, pp. 1051–1101, 2006.

[15] M. Freed, Adrian, and Wright, "Open Sound Control." Centre for New Music and Audio Technologies.

[16] "MAX/MSP." Cycling 74.

[17] "Live." Ableton.

[18] P. M. Todd and G. M. Werner, *Frankensteinian Methods for Evolutionary Music Composition*. Cambridge, MA: MIT Press/Bradford Books, 1999.

FACTORS INFLUENCING VOCAL PITCH IN ARTICULATORY SPEECH SYNTHESIS: A STUDY USING PRAAT

Sivaramakrishnan Meenakshisundaram
SASTRA University, Thanjavur, India & Interdisciplinary Centre for Computer Music Research, Plymouth University, Plymouth, United Kingdom

Eduardo R. Miranda
Interdisciplinary Centre for Computer Music Research, Plymouth University, Plymouth, United Kingdom

Irene Kaimi
School of Computing, Electronics & Mathematics Plymouth University, Plymouth, United Kingdom

ABSTRACT

An extensive study on the parameters influencing the pitch of a standard speaker in articulatory speech synthesis is presented. The speech synthesiser used is the articulatory synthesiser in PRAAT. Categorically, the repercussion of two parameters: Lungs and Cricothyroid on the average pitch of the synthesised sounds are studied. Statistical analysis of synthesis data proclaims the extent to which each of the variables transforms the tonality of the speech signals.

Keywords: Articulatory Synthesis, PRAAT, Vocal Pitch, Cricothyroid, Lungs.

1. INTRODUCTION

A speech synthesiser, outlined in terms of articulatory parameters is a model that incorporates a human vocal tract system. The production of speech sounds using this model is known as articulatory synthesis. Exploring robust synthesis techniques that can substitute concatenative synthesis has recently become an established field of study. This is due to the limitations that concatenative synthesis imposes on modifying the expressivity of sounds in real time. Articulatory synthesis has been contemplated to have the greatest potential out of the contemporary synthesis techniques [1,2]. Perception of anatomy and physiology of a human vocal system is statutory to work with articulatory synthesis. A human vocal tract system can be thought of as a resonating acoustic structure with temporal properties [3]. The entire vocal apparatus is treated as an air filled cavity with walls that may be analogised to adjustable mass-spring systems [4].

The state of the articulatory synthesiser at a particular juncture can be represented by the position of the organs of speech. The time-varying variables of the model simulate a quasistatic speech event that may be represented as a sequence of stationary responses of the model, each corresponding to a particular configuration of the articulatory parameters.

Articulatory parameters have advantage therein they describe the system that produces the sound instead of the results of that method [1]. However, the irregular form of the vocal tract and temporal properties of the system increase the complexity of modelling.

This project employs the articulatory synthesiser in PRAAT, developed by Paul Boersma and David Weenink. PRAAT is a sophisticated platform for analyzing, synthesizing and manipulating speech [5]. PRAAT comes with its own scripting language and an ideal user interface for analysis and production of speech signals.

The articulatory synthesiser in PRAAT offers 29 degrees of freedom, each typifying an organ/articulatory parameter of the vocalisation system. These parameters of the synthesiser are excited by passing numerical values as input. The physical model in PRAAT provides appreciable realism and naturalness of the sounds synthesised, increasing the prospects of implementation of text-speech systems based on articulatory synthesis in the near future.

This paper presents a comprehensive study on the factors influencing vocal pitch in articulatory synthesis using PRAAT. The two main parameters that alter the pitch of a human voice are air flow from the Lungs and the Vocal Fold Tension[4]. The cynosure is on the Cricothyroid parameter of the model that is related to the Vocal Fold Tension. This variable is extrapolated beyond the nominal range to observe for changes in the pitch of the sounds synthesised. The results acquired are motivational for further explorations in this discipline.

2. METHODOLOGY

PRAAT features a sophisticated synthesiser that is capable of producing realistic vocal sounds of great interest to composers and artists [6]. For this exploratory measure, the physical model is constrained to 6 parameters to reduce complexity. The model is configured to a standard speaker with two tubes in the glottis. In PRAAT parlance, the Artword Object that encloses all the muscle components. These components can be excited either by directly modifying the Artword or by using the scripting tool in PRAAT. The Artword can be created from the main menu or by using the PRAAT script. The entire set of operations is done on an "A" vowel sound synthesised using the model. In theory, the parameters of the articulatory synthesiser can vary from -1.0 to +1.0, but in most of the cases we employ 0.0 as the starting point [6].

2.1 Excitation of Parameters

Vocal fold oscillation eventuates in an event of speech. This process can be explained with the Myoelastic-Aerodynamic theory. According to the theory, Bernoulli forces create a closed airspace below the glottis by sucking the vocal folds together. Once the subglottal pressure is high enough, the folds are blown outward causing phonation.

Under conditions of zero vocal fold collision and idealized flow in the glottis the intraglottal pressure can be written as [7],

$$P_g = \left(1 - \frac{a_2}{a_1}\right)(P_s - P_i) + P_i \quad (1)$$

where a_1 and a_2 are the cross-sectional areas at the entry and exit points of the glottis respectively, P_s is the subglottal pressure and P_i is the input pressure to the vocal tract. The term $(P_s - P_i)$ represents the transglottal pressure. Eqn. 1 clearly illustrates the process of vocal fold oscillation resulting in phonation.

This preceding section elucidates the configuration of the muscle parameters enfolded in the Artword, with numerical values that can produce a significant utterance for analysis. This is a time domain approach of simulating the dynamic properties of a human vocalisation system.

The Lungs parameter in the model produces the necessary air pressure to cause phonation. This parameter can be set to attain values between -0.5 and +1.5: where -0.5 represents the maximum volume of air exhaled by the speaker and the value +1.5 represents the maximum volume of air inhaled [2].

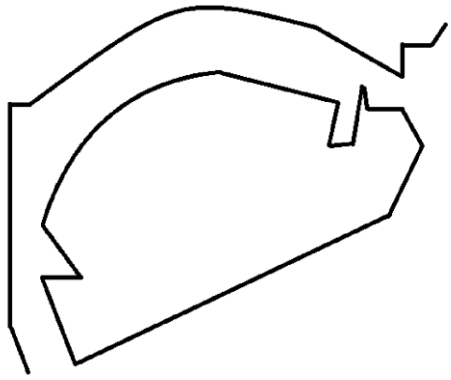


Figure 1. Expiration in PRAAT's articulatory synthesiser

Figure 1 shows the Exhale operation of the articulatory synthesiser in PRAAT. The Exhale operation can be implemented by reducing the equilibrium width of the Lungs. Adduction of vocal folds is due to the Interarytenoid muscle that connects the two paired, pyramidal Arytenoid cartilages [8]. It consists of two components: the Transverse Arytenoid and the Oblique Arytenoid. These muscles run horizontally across each other forming a shape of letter 'X". This parameter of the synthesiser is set to 0.5 throughout the utterance to produce a normal voicing. Masseter is a thick rectangular muscle that carries out mastication [9]. This criterion of the model enables opening and closure of the jaw. For an "A" vowel to be synthesised, the jaw should be open. Hence the variable is configured to have a value of -0.4 throughout the utterance. Similarly, a value of 0.5 can be set for complete jaw closure.

The Hyoglossus, one of the extrinsic muscles of the tongue is quadrangular in shape in the lower two-thirds and radiates into a fan shaped structure in upper one-third [10]. This muscle is responsible for depression and retraction of the human tongue and is a cardinal muscle in singing. In order to produce an "A" sound, this muscle variable is excited with a value of +0.4 throughout the utterance.

Cricothyroid is a muscle that influences the pitch of the sound produced in a human vocal tract by altering the tension and the length of the vocal folds [4,6]. This muscle stretches forwards and backwards to modulate the pitch. Akin to tightened guitar string, the stretched Cricothyroid muscle produces a high-frequency sound. The Cricothyroid parameter of the synthesiser is of great interest as recent experiments have shown the muscle's direct relation to the fundamental frequency of the human voice [11]. This parameter is excited with values between 0 and 4 to observe for changes in the vocal pitch of the synthesiser.

PRAAT has a great advantage in producing sounds such as nasals to a substantial extent of realism. The LevatorPalatini muscle parameter of the synthesiser is responsible for opening and closure of the Velopharyngeal port. Oral phonemes require the port to be closed and the nasal phonemes are produced by nasal resonance, which is accomplished by opening the Velopharyngeal port and allowing the air to pass through the nasal cavity. This operation in PRAAT can be achieved by exciting the LevatorPalatini variable. In this project, as an "A" vowel is synthesised, a value of +1.0 is assigned to the parameter for enabling complete closure of the Velopharyngeal port. A value of 0.0 enables opening of the port.

2.2 Synthesis

PRAAT is highly flexible that it is adequate to just specify the discrete settings of the parameters. It automatically introduces the values in between the discrete levels. A sustained phonation is achieved by setting the duration of the utterance to 1.5s. This gives a wider window to analyse the behaviour of the pitch throughout the utterance. The time-varying variables of the model are excited at discrete time levels that are within the length of the phonation.

Every variable of the Artword can be excited through PRAAT script. For example the statement: Set target: 0.03, -0.1, "Lungs", initializes the Lungs parameter with a value of -0.1 at the time instant 0.03. All other parameters are excited in similar fashion.

Once all the variables are initialized, the Artword along with the Speaker properties is synthesised to a sound. PRAAT offers 9 different options (Sampling frequency, Oversampling factor, Width 1, Width 2, Width 3, Pressure 1, Pressure 2, Pressure 3, Velocity 1, Velocity 2, and Velocity 3) to synthesise the sound. The sound synthesised is directly related to the muscles that are configured in the model.

All results presented in this paper are based on the "Average Pitch" of the sounds, as the pitch tends to vary

along the utterance. Not all parameter configurations result in a normal voicing. Some of the configurations tend to produce voiceless sounds that don't have a definite pitch. There may also be breaks in some of the sounds synthesised. It is difficult to say if these breaks are related to the realistic phenomena of the model [4]. Another noteworthy characteristic of the model is that the sounds relating to a particular configuration, when re-synthesised after a certain interval of time result in different tonal and spectral properties, manifesting the imperfect nature of the physical model in PRAAT. Aforementioned qualities attribute to the unexploited nature of articulatory speech synthesis in robust applications.

3. RESULTS

In order to get an insight into the functioning of the model, a very large set of sound samples is required. Each of these sound samples is related to a particular configuration of the Artword parameters. Four of the six parameters are kept constant throughout the experiment. For this experimental study, the model is simulated with different combinations of air flow and vocal fold tensions along with the other four static parameters. This results in 450 different speech sounds of same phonation length with different tonal and spectral properties. These sounds are statistically analysed to find the effect of these variables on the average pitch of the speaker to which the synthesiser is configured.

3.1 Histogram

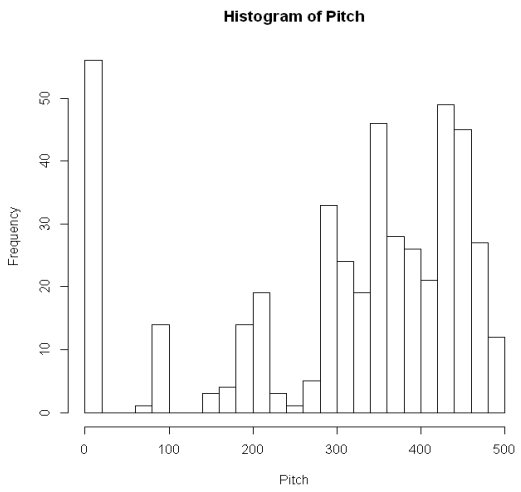


Figure 2. Histogram of Vocal Pitch (Average) in Hz data corresponding to different configuration of Lungs and Cricothyroid

The Histogram of the variable pitch is first plotted. As Figure 2 suggests that the distribution of pitch is not symmetric (due to the undefined pitch levels). This suggests that the normality of the variable cannot be assumed. Then a formal statistical test for normality (a Shapiro-Wilk test) is done. The p-value of the test is 0.000, hence the null hypothesis of normality is rejected. As a result, parametric tests cannot be performed hence, non-parametric methods to assess association of pitch with the available covariates are employed.

3.2 Effect of Exhale Levels on Resultant Pitch

In PRAAT, the Lungs parameter has a greater impact on the amplitude of the sounds synthesised. In addition to this, it also influences the pitch of the synthesiser on a smaller scale. This effect should be taken into consideration as they tend to stimulate the Cricothyroid. From Figure 3, it can be seen that the distribution of the pitch is similar for all six levels of Exhale. The median values of the pitch in the six groups (shown as the solid lines inside the boxplots) are approximately at the same value. This suggests that there is minimal Exhale effect on pitch. Then a Kruskal-Wallis test is also performed to statistically assess the Exhale effect on pitch. This gives a p-value=0.798, hence the null hypothesis of the test, that there is no much difference in the median values of the average pitch for the six levels of Exhale, cannot be rejected, which suggests that there is no statistical evidence of an Exhale effect.

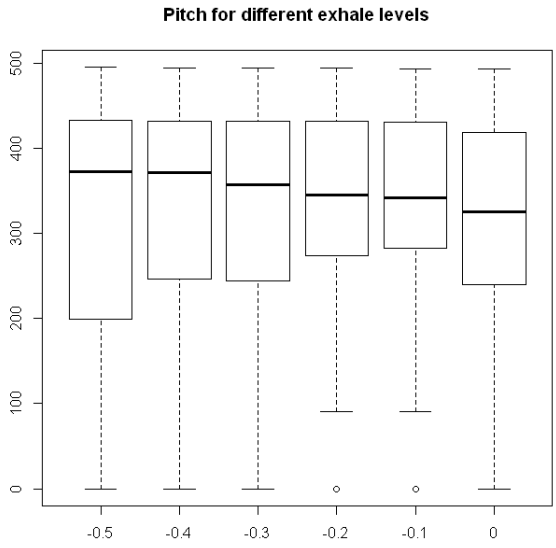


Figure 3. Vocal Pitch (Average) in Hz plot for six levels of Exhale configuration of the Lungs parameter of the synthesiser

3.3 Effect of Cricothyroid Levels on Resultant Pitch

Figure 4 indicates that the distribution of the pitch differs for the six levels of Cricothyroid. For example, at level 0, pitch takes a wide range of values, whereas, at level 4 the range of values of pitch is very limited. The median values of pitch in the six groups (shown as the solid lines inside the boxplots) appear to be very different. For example, at level 0 the median is close to 0, whereas at level 5 the median Pitch value is around 450. This suggests that there may be a Cricothyroid effect on pitch. Then a Kruskal-Wallis test is done to statistically assess the Cricothyroid effect on pitch. This gives a p-value=0.000, hence there is overwhelming evidence to reject the null

hypothesis of the test, that there is no difference in the median values of pitch for the five levels of Cricothyroid. This suggests that there is a strong statistical evidence of a Cricothyroid effect.

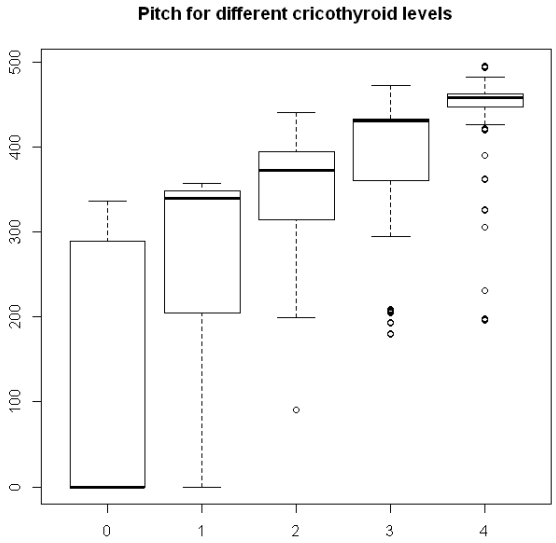


Figure 4. Vocal Pitch (Average) in Hz plot for five levels of the Cricothyroid parameter of the synthesiser

3.4 Effect of Inhale Levels on Resultant Pitch

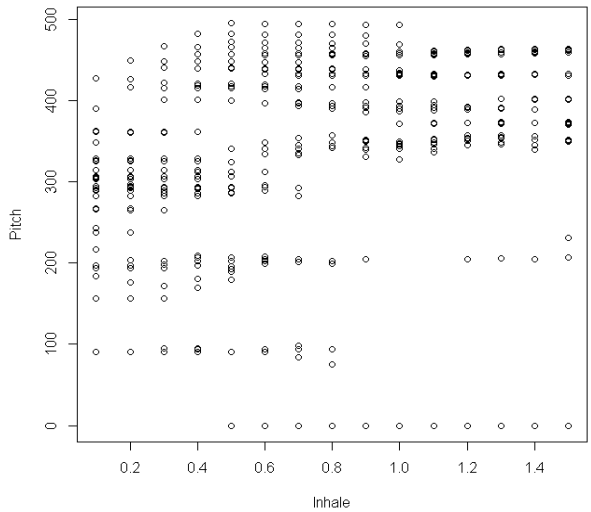


Figure 5. Vocal Pitch (Average) in Hz plot for seven levels of Inhale configuration of the Lungs parameter of the synthesiser

Figure 5 does not reveal any obvious relationship between pitch and Inhale. This suggests that there is probably no Inhale effect on pitch. A Kendall tau rank correlation coefficient test is performed to statistically assess the Inhale effect on pitch. This gives a low p-value<0.001, hence there is an evidence to reject the null hypothesis of the test, that the correlation between pitch and Inhale is equal to 0. However, the size of the correlation coefficient is 0.19 (very low, compared to 1 which would indicate perfect linear relationship), which suggests that there is no association between pitch and Inhale.

3.5 Non-Parametric Model

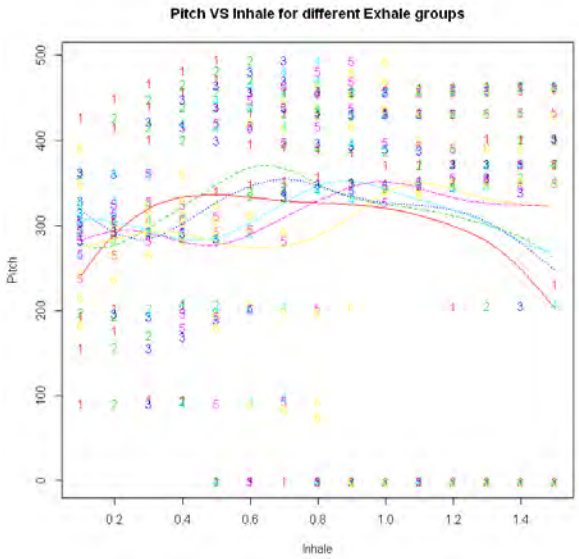


Figure 6. Vocal Pitch (Average) in Hz Vs Inhale plot for different Exhale groups

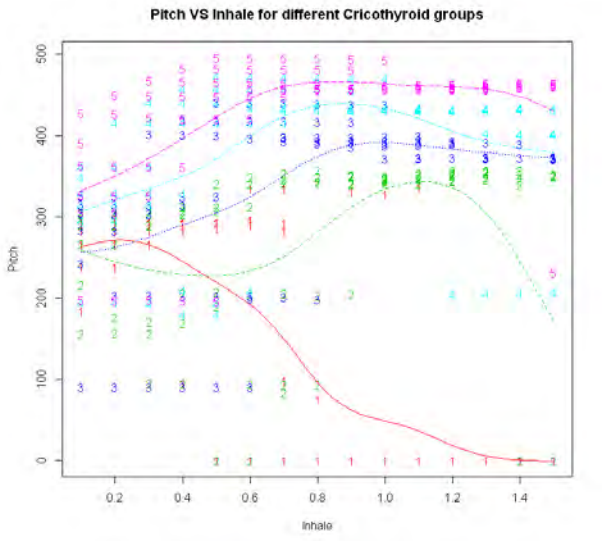


Figure 7. Vocal Pitch (Average) in Hz Vs Inhale plot for different Cricothyroid groups

Finally, the non-parametric equivalent of an ANCOVA model is fitted to the data (in which all the available covariates: Inhale, Exhale and Cricothyroid are included) and their significance and how they affect the response pitch is assessed.

As shown from Figures 6 and 7, there is no difference between different exhale groups, whereas different Cricothyroid groups differ in how pitch and inhale are associated. The results confirm that there is no association between Inhale and pitch, and that the pitch medians are the same for all six levels of exhale. However, there is a difference in the median pitch of different Cricothyroid levels. Multiple comparisons provide evidences of statistical

differences between all pairs of Cricothyroid levels (0-1,0-2,0-3,0-4,…,4-1,4-2,4-3).

4. CONCLUSIONS AND FUTURE WORK

A detailed analysis of the repercussion of articulatory parameters on vocal pitch has led to the following conclusions.

- Not all parameter configurations result in a normal voicing. This evinces the realistic phenomena of the articulatory model in PRAAT.
- Inhale pressure levels of the Lungs do not have any effect on the intrinsic vocal pitch of the sounds synthesised using the model.
- The resultant pitch medians of different Exhale levels are approximately at the same range. Thus no statistically significant effect of the Exhale pressure levels on the tonality of the speech sounds synthesised can be identified.
- Divergence in pitch medians for different levels of Cricothyroid, reinforces the fact that the vocal cord tension is directly related to the intrinsic vocal pitch of the speech signals. Statistical studies also arrive to a conclusion that the Cricothyroid has a solid role in altering the pitch of the synthesiser.

Future work will go towards exploring other articulatory parameters of the model that transform the tonality of the speech sounds. To initiate, the Thyroarytenoid and the Vocalis parameters of the model will be studied extensively along with the other six parameters used in this experiment. Further explorations will lead the way to discover parameter configurations for different vowels and consonants, transition between individual sounds using the model.

5. REFERENCES

[1] C. H. Shadle and R. I. Damper, "Prospects for articulatory synthesis: A position paper," 2002.

[2] J. Drayton and E. Miranda, "Towards an Evolutionary Computational Approach to Articulatory Vocal Synthesis with PRAAT," in *Evolutionary and Biologically Inspired Music, Sound, Art and Design*, ed: Springer, 2015, pp. 62-70.

[3] C. Wu and Y.-F. Hsieh, *Articulatory speech synthesizer*: University of Florida, 1996.

[4] P. Boersma, "Functional phonology," ed: The Hague: Holland Academic Graphics, 1998, pp. 31-63, pp. 113-140.

[5] P. Boersma and V. van Heuven, "Speak and unSpeak with PRAAT," *Glot International*, vol. 5, 2001, pp. 341-347.

[6] E. Miranda, *Computer sound design: synthesis techniques and programming*: Taylor & Francis, 2012, pp. 137-152.

[7] M. A. Redford, *The handbook of speech production*: John Wiley & Sons, 2015, pp. 34-58.

[8] C. A. Rosen and B. Simpson, *Operative techniques in laryngology*: Springer Science & Business Media, 2008, pp. 3-8.

[9] T. Van Eijden, "Jaw muscle activity in relation to the direction and point of application of bite force," *Journal of dental research*, vol. 69, 1990, pp. 901-905.

[10] S. Abd-El-Malek, "Observations on the morphology of the human tongue," *Journal of Anatomy*, vol. 73, 1939, pp. 201-210.

[11] D. Erickson, T. Baer, and K. S. Harris, "The role of the strap muscles in pitch lowering," *Vocal fold physiology: Contemporary research and clinical issue*,1983, pp. 279-285.

TOWARDS A VIRTUAL-ACOUSTIC STRING INSTRUMENT

Sandor Mehes, Maarten van Walstijn, Paul Stapleton
Sonic Arts Research Centre
School of Electronics, Electrical Engineering and Computer Science
Queen's University Belfast
Belfast, UK
{smehes01,m.vanwalstijn,p.stapleton}@qub.ac.uk

ABSTRACT

In acoustic instruments, the controller and the sound producing system often are one and the same object. If virtual-acoustic instruments are to be designed to not only simulate the vibrational behaviour of a real-world counterpart but also to inherit much of its interface dynamics, it would make sense that the physical form of the controller is similar to that of the emulated instrument. The specific physical model configuration discussed here reconnects a (silent) string controller with a modal synthesis string resonator across the real and virtual domains by direct routing of excitation signals and model parameters. The excitation signals are estimated in their original force-like form via careful calibration of the sensor, making use of adaptive filtering techniques to design an appropriate inverse filter. In addition, the excitation position is estimated from sensors mounted under the legs of the bridges on either end of the prototype string controller. The proposed methodology is explained and exemplified with preliminary results obtained with a number of off-line experiments.

1. INTRODUCTION

Synthesis by physical modelling is designed as the ultimate methodology for digital simulation of real-world instruments [1–4]. The key difference with sample-based approaches is that the synthesis algorithm captures and parameterises the physical behaviour rather than the signal output. Hence in principle, *virtual-acoustic instruments* can be designed on this basis that are similar to real-world acoustical instruments in the way they sonically respond to player actions and afford performance nuances. However while significant advances have been made over the past few decades regarding numerical modelling of musical instruments, relatively little progress has been made so far in terms of real-time control of the resulting algorithms.

It is worthwhile noting at this point that the problem of synthesis control has been much more widely investigated as a *gestural mapping* problem (see, e.g. [5–8]). Generally this concerns a more free approach to the design of new, computed-based musical instruments, usually without specific consideration of the constraints and characteristics of acoustics instruments, and often instilling a more loose coupling between the player and the instrument. The mapping problem does in fact not exist in the same way when using physical modelling synthesis, since the model parameters generally have direct counterparts in the real world. This suggests that the mapping can in principle be replaced by a *direct routing* between a real-world controller and a virtual-domain sounding system (i.e. the physical model algorithm), with the interface dynamics directly inherited from the modelling process [9]. Fig. 1a illustrates this concept schematically. Following several earlier studies, the physical model is represented here in terms of its block decomposition into an "exciter" (i.e. an excitation object such as a bow or a finger) and a "resonator" (i.e. a vibrating structure such as a string or a membrane). Traditionally, the exciter, the resonator, and their interaction are all modelled (i.e. existing in the virtual domain), with as-

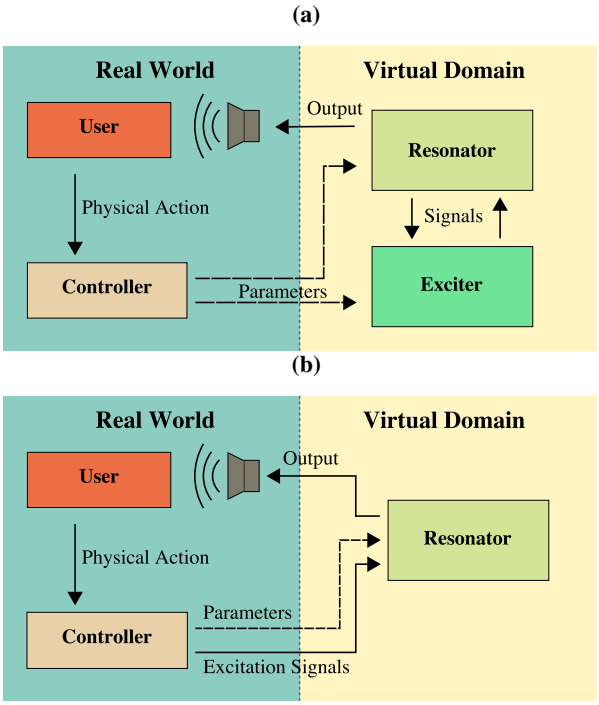


Figure 1: (a) A 'conventional' configuration for real-time control of a physical model. (b) An alternative configuration in which the excitation signals are generated with the controller.

sociated parameters that are to be controlled by the player. For example, playing a bowed string model involves real-time adjustment of the bow parameters (e.g. bow speed, bow force) as well as of the string parameters (e.g. finger stopping position). One of the main challenges in realising such a conventional physical model configuration arises from the high computational costs involved in precise modelling of all of the physical mechanisms involved (see, for example, the case of a two-polarisation bow-string model [10]).

Leaving aside such efficiency concerns, the remaining challenge focuses on controller design, which invariably involves perpending the larger scope of multi-modal interaction, i.e. also including forms of haptic and visual feedback. This topic has been extensively investigated in the past few decades within the sound and computing community as well as in the wider realm of human-computer interaction, and has resulted in various strands of related controller design concepts, including those based on *natural* [11], *tangible* [12, 13], *embodied* [12], *enactive* [13] and *effortful* [14] interaction. The current paper is partly inspired by these concepts, and in alignment with them seeks a sensor configuration that minimises its interference with the instrumentalist's actions.

In light of such interaction design criteria, Berdahl and Smith [15] proposed a slightly different configuration for physical model control, which leaves the exciter part in the real-world domain (see Fig. 1b). In this arrangement, the physical form of the controller resembles the main vibrating element of the simulated instrument. In the case studied in [15], the player is presented with a (silent) controller interface with two strings, one of which is damped and excited in the usual ways (plucking, striking, etc.) in order to drive a physics-based string resonator model, while the other controls the pitch.

A key technical challenge that arises in this approach is to ensure that the interface dynamics are captured in appropriate form for driving the virtual-domain sound resonator, which boils down to 'clean extraction' of the relevant excitation signal(s). That is, the controller should comprise real-time sensing/processing of signals in order to obtain an equivalent of the signal(s) normally flowing from the exciter to the resonator. For example, in the case of percussive string excitation, the signal that is most suited to excite a virtual string model is the actual force signal exerted by the player on a (strongly damped) string, with any possible distortion by the setup (e.g. coloration by the sensors) removed as much as is feasible. In addition, the envisaged application to performance requires a high-quality audio, low-noise excitation signal. In [15] this is addressed by using an electric guitar as the tangible interface, fitted with undersaddle piezoelectric pickups to sense the string vibrations. The piezos are more suitable than bridge pickups due to the inherently nonlinear characteristics of the latter. The (augmented) use of the electric guitar as the physical controller has consequences regarding the type of control and idiom a performer is invited to engage with. The probability of players and virtual-acoustic instrument designers reinventing string playing in a way that genuinely expands

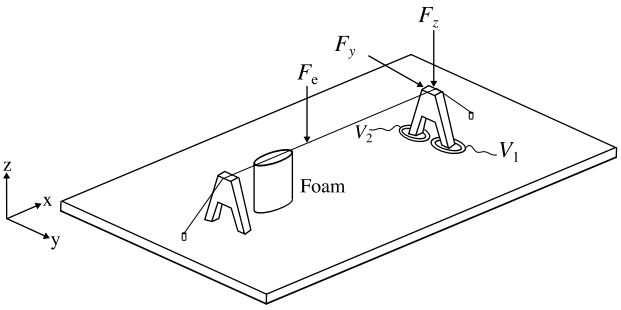


Figure 2: Setup of the prototype string controller. Under each leg of the bridges (a ,b) piezoelectric disks are placed, generating voltages (V_1, V_2) resulting from the pressure of the vibrating string on the bridges.

artistic practices is further reduced if the string model parameters are restricted to a range that produces sounds that are close to the sonic palette of conventional guitar sounds.

This paper is similarly motivated, but takes a different approach by moving away from using the electrical guitar and associated commercial piezo pickups for the prototype string controller. The main purpose is to keep more design flexibility, which aligns with the longer-term aim of more adventurously exploring the acoustic affordances of virtual-acoustic string instruments. The principal technical novelty of the work presented here is that specific attention is given to removing the characteristics of the sensor via linear filtering and pre-calibration. In addition, a method for estimating the excitation position (from the same sensor data) is proposed. The excitation type is restricted largely to percussive styles (i.e. resulting from short interactions between the string and a finger/object), since the technical challenges involved in separating the 'exciter' from the 'resonator' are considerably more complex for fully sustained excitation (i.e. string bowing).

The remainder of the paper is structured as follows. The prototype string controller is presented in Section 2, including the signal processing used for estimating the player force signal and the excitation position from the vibrations sensed at the instrument bridge. Section 3 then gives a summary of the modal synthesis string resonator model and its implementation, followed by the exposition of a few exemplifying preliminary off-line results in Section 4.

2. A PROTOTYPE STRING CONTROLLER

2.1 Experimental Setup

A string is stretched over two bridges mounted on a wooden support platform, as depicted in Figure 2. Currently the bridges of a Guzheng (a Chinese string instrument [16]) are used. A piece of foam is placed close to the left bridge with the purpose of damping the vibrations of the string, as such subduing multiple round-trip wave reflections. Assuming linear wave propagation and neglecting string stiffness and damping this means that - apart from at very low frequencies - any transversal force seen at the bridge furthest from the foam is approximately equal to a delayed version of the force wave travelling towards bridge generated when

the player excites the string

$$F(t) \approx F_{\mathrm{e}} \left(t - \frac{L - x_{\mathrm{e}}}{c} \right), \qquad (1)$$

where $L = 0.460$m is the string length between the two bridges, x_{e} is the excitation position, and c is the velocity at which transversal waves travel along the string. Hence the foam placement allows directly extracting the excitation force from $F(t)$, be it with latency $\tau = (L - x_{\mathrm{e}})/c$.

The setup also features foam strips that are glued to the bottom of the support platform to absorb external vibrations that could corrupt the signal. To sense the vibrations at the bridge, a piezoelectric disk (PD) is positioned under each of its two legs. The analogue signal routing for these sensors contains a high-pass circuit which helps attenuating the DC component (including any signal 'drift' that would be detrimental to any further processing). The piezo sensor signals are digitally captured with an NI USB-6215 data acquisition platform. The final stage of the processing chain is a computer [1] for both the parameter estimation and the real-time implementation of the string resonator model.

2.2 Excitation Force Signal Estimation

Referring again to Fig. 2, the strategy here is to set up an inferential sensing system to estimate the vertical (F_z) and horizontal (F_y) forces exerted on the bridge by the string. Under the assumption of linear behaviour of both the bridge and the sensors, the PD signals — denoted here as $V_i(t)$, where $i = 1, 2$ — are simply filtered versions of the force signals. Under vertical force excitation, the frequency-domain relationships then are:

$$V_i(\omega) = G_i(z) F_z(\omega), \qquad (2)$$

where $G_i(z)$ is the corresponding transfer function, encapsulating the characteristics of the bridge, $F_z(\omega)$ is Fourier transform the vertical force excitation on the bridge and $V_i(\omega)$ is the Fourier transform of the piezo signal. In order to estimate the vertical force signal, the relationships in eq (2) must be inverted. Fig. 3 illustrates how this can be realised in the time domain, using calibration filters $C_i(z)$ that approximate the inverses of the transfer functions $G_i(z)$ (see the next Section for the design of these filters).

For vertical forcing of the bridge, the string pushes downwards on the two legs simultaneously, resulting in in-phase piezo signals. Therefore summing the filtered signals and multiplying by $1/2$ gives an estimation of the vertical force component by averaging; this is realised with the upper arm of the signal processing diagram in Fig. 3. On the other hand, with a horizontal force impact on the bridge one leg of the bridge is lifted up while the other leg is pushed down resulting in signals that are out of phase with each other. Therefore a horizontal force estimate $\hat{F}_y$ can be obtained by subtracting the filtered signals from the PD. The difference in exciting in the vertical as opposed to the horizontal plane can in itself be considered as a filtering

[1] iMac 2.8GHz quad-core Intel Core i5,16 GB of 1867MHz LPDDR3 onboard memory

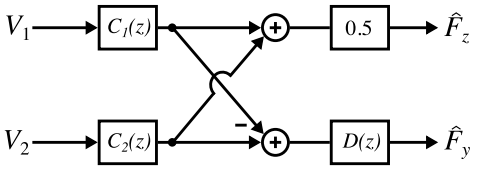


Figure 3: Signal diagram for the force estimation. The voltages (V_1, V_2) are generated by the piezoelectric disks processed by the calibration filters, $C_1(z)$ and $C_2(z)$. The sum and difference give estimation of the bridge forces ($\hat{F}_z$, $\hat{F}_y$). Filter $D(z)$ can be added for fine tuning the $\hat{F}_y$ estimation.

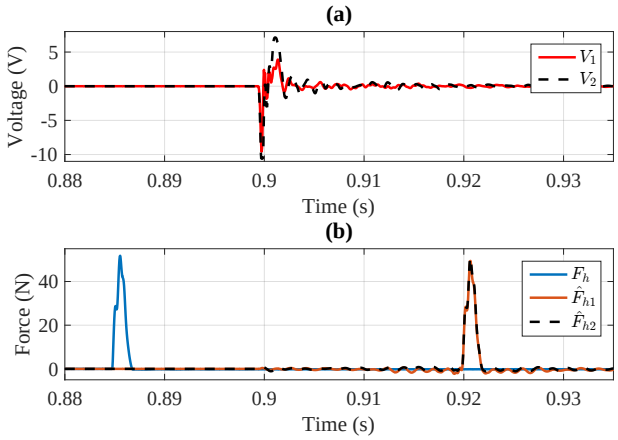


Figure 4: Voltages (a) generated by piezoelectric disks placed under the bridge. Force measured with an impact hammer (b) applied at the bridge (blue) and estimations from V_1 (red) and V_2 (black, dashed).

effect, and can thus potentially be modelled with an additional filter $D(z)$ for increased accuracy for $\hat{F}_y$; this extension has not been yet realised or tested within the project though, and instead the difference signal is currently taken directly as a measure of the vertical bridge force.

2.3 Identification of the Calibration Filters

In order to estimate the forces exerted on the bridge from the sensed PD signals as described above, digital filters $C_i(z)$ that approximate the inverses of $G_i(z)$ are required. To obtain such filters, a pre-calibration experiment is carried out by measuring the force impact on the bridge when it is hit by an impact hammer [2] from above, and simultaneously sensing the PD signals. An example set of measurement signals is plotted in Figure 4. An optimum inverse filter can then be designed for each piezo through various means; here adaptive filtering methods [17] are applied to provide a first, preliminary result, in the form of an FIR filter. In particular the recursive least squares (RLS) algorithm is useful in this case because of its relatively high robustness against input signal characteristics. A suitable set of input/output training signals is created by first convolving the hammer and piezo signals with a white pseudo-noise signal, ensuring that the input signals are of sufficient

[2] Dytran Dynapulse 5800B4

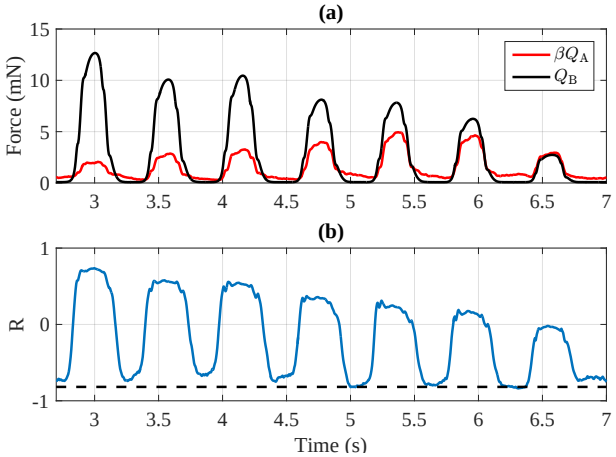


Figure 5: Position estimation signals obtained when striking the string successively at positions $x_e = 373.5$, 353.0, 332.5, 312.0, 291.5, 271.0, and 250.5mm, respectively. (a) Short-time rectified average of the two estimated force signals. (b) The corresponding evolution of the variable $R(t)$. The dashed line indicates the 'default value' $(1-\beta)/(1+\beta)$ which it approximately attains in the absence of excitation.

length to train the RLS algorithm. The error signal (the difference between the target and estimate signal) is defined allowing for a small time delay of the FIR filter's impulse response; the results presented used at a sampling rate of 51.2kHz. As seen in Figure 4(b), filtering $V_i(t)$ through the calibration filters this way results in an accurate estimation of $F_z(t)$ from each PD (and thus also from the averaged signal obtained with the upper arm of the signal diagram in Figure 3.

Note that the equalisation that is carried out by passing the piezo signal though the calibration filter affects both amplitude and phase characteristics. As a result, sharp force pulses are reconstructed by the calibration filters from piezo signals that are more 'smeared' over time. This means that transient-rich detail in the excitation signal (arising from the player's interaction with the string) is exposed more sharply in the final audio signal than if the piezo signal were to be passed straight to the resonator model.

A drawback of the adaptive filtering approach is that using long FIR filters can be computationally demanding, making it less suitable for real-time application. A more efficient approach is possible though, by first extracting the dominant modes of the bridge and implementing these separately as second-order resonance filters [18].

2.4 Excitation Position Estimation

In order to estimate the position at which the string is excited by the player, piezo sensors are also fitted under the legs of the other bridge. One approach would be to determine the time difference between the signals arriving at the bridges, with pulses due to plucks positioned closer to the right-hand bridge (B) arriving earlier at that bridge than at the left-hand side bridge (A). However the presence of the foam, which causes temporal smearing of wave pulses travelling towards bridge (A) complicates this approach. Instead the estimation approach taken here is based on determining the short-time RMS-like signal averages $Q_A(t)$ and $Q_B(t)$ at the two bridges. For each bridge, this signal is calculated by first applying a first-order low-pass filter to the estimated force, then applying signal rectification (by taking the absolute value the signal), and finally applying a smoothing (moving average) filter. Fig. 5(a) shows an example set of signals when the string is struck successively at seven different positions. The two signals are then used in the calculation of the dimensionless quantity

$$R(t) = \frac{Q_B(t) - \beta Q_A(t)}{Q_B(t) + \beta Q_A(t)}, \tag{3}$$

where β is a constant compensating for the foam damping (here $\beta = 10$ is used); the damping by the foam is approximately constant within the low-frequency band that is effectively used in the signal calculation. Fig. 5(b) shows how $R(t)$ varies with striking position. In periods of no excitation, the value of R is approximately $(1-\beta)/(1+\beta)$ due to the noise on the signals from which $Q_A(t)$ and $Q_B(t)$ are calculated. More generally, $R(t)$ relates to $x_e(t)$ through a static nonlinear mapping $R = \mathcal{G}(x_e)$; this map can be obtained by a further pre-calibration measurement involving multiple plucks at a range of positions along the string followed by a curve fitting. Fig. 6 shows an example result of this process, in which we retrieved the map in its simplest possible form, i.e. a straight line. This procedure prepares for the estimation in real-time of the excitation positions in the range $[L/2 - L]$ employing the inverse of the obtained mapping, i.e.

$$x_e(t) = \mathcal{G}^{-1}(R(t)). \tag{4}$$

The inconsistencies in R seen in Fig. 6 for any of the string excitation positions are due to (a) extraneous measurement signals and (b) string vibrations resulting from effectively stopping the string with the plucking object (hence setting up wave roundtrips over the string portion between $x = x_e$ and $x = L$. The latter problem is unavoidable to certain extent, but the former can be alleviated by improved signal conditioning.

3. STRING RESONATOR MODEL

3.1 String Model

Transversal string vibrations, taking into account non-idealities such as stiffness and damping, can be described with the partial differential equation [3, 4]

$$\rho A \frac{\partial^2 y}{\partial t^2} = T \frac{\partial^2 y}{\partial x^2} - EI \frac{\partial^4 y}{\partial x^4} - \gamma(\beta) \frac{\partial y}{\partial t} \quad + \mathcal{F}_e(x,t), \tag{5}$$

in which ρ, A, T, E, and I are mass density, cross-sectional area, tension, Young's modulus, and moment of inertia, respectively, and where $y(x,t)$ denotes the transversal displacement at axial position x and time t. Given that the support platform is thick, strong, and heavy, simply supported boundary conditions can be assumed:

$$y(x,t)\Big|_{x=0,L} = 0, \quad \frac{\partial^2 y}{\partial x^2}\Big|_{x=0,L} = 0. \tag{6}$$

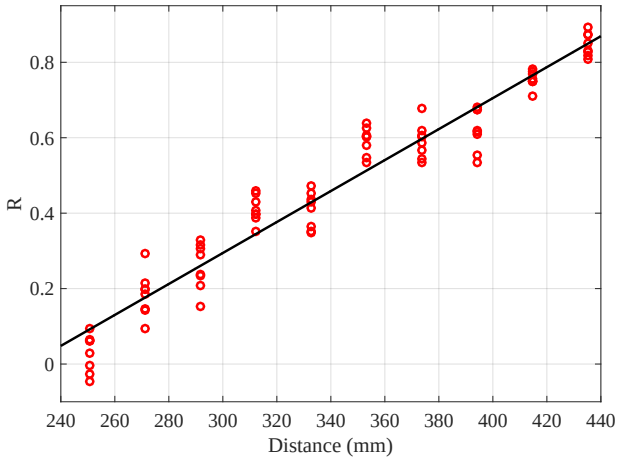


Figure 6: Mapping between R and x_e. The circles indicate individual measurement data points, and the line is a least-squares fit to the data.

Frequency-dependent string damping is incorporated here by defining the parameter $\gamma(\beta)$ in (5) as:

$$\gamma(\beta) = 2\rho A\Big[\sigma_0 + \left(\sigma_1 + \sigma_3\beta^2\right)|\beta|\Big], \tag{7}$$

where β is the wave number and $\sigma_{0,1,3}$ are physically-motivated fit parameters. The external excitation is restricted here to a single point, i.e.

$$\mathcal{F}_\mathrm{e}(x,t) = \delta(x_\mathrm{e})F_\mathrm{e}(t), \tag{8}$$

where $F_\mathrm{e}(t)$ and x_e represent the force signal and excitation position, both of which are directly obtained from the controller within the proposed approach. An appropriate audio signal can be obtained by calculating the termination force at $x = L$:

$$F_\mathrm{T}(t) = -T\frac{\partial y}{\partial x}\Big|_{x=L} + EI\frac{\partial^3 y}{\partial x^3}\Big|_{x=L}, \tag{9}$$

and inputting this to a body filter, such as those designed in [18]. Alternatively, the string velocity at a pick-up position x_p can serve as a sound output signal.

3.2 Modal Synthesis

The solution of (5) can be expressed as a superposition of the normal modes of the string (indexed with i):

$$y(x,t) = \sum_{i=1}^{M} v_i(x)\,\bar{y}_i(t), \tag{10}$$

where $\bar{y}_i(t)$ denotes the mode displacement and $v_i(x) = \sin(\beta_i x)$ is the corresponding mode shape (spatial eigenfunction) for the boundary conditions given in (6), with $\beta_i = i\pi/L$. Substitution of (10) into (5), then multiplying with $v_i(x)$ and applying a spatial integral over the length of the string yields that the dynamics of each of the modes is governed by:

$$m\frac{\partial^2 \bar{y}_i}{\partial t^2} = -k_i\bar{y}_i(t) - r_i\frac{\partial \bar{y}_i}{\partial t} + v_i(x_\mathrm{e})F_\mathrm{e}(t), \tag{11}$$

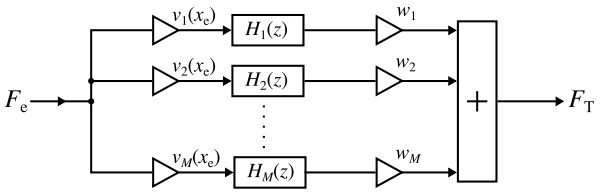


Figure 7: Signal diagram for the modal synthesis algorithm. The output weights w_i are computed with (13) or alternatively set as $v_i(x_\mathrm{p})$ for velocity pickup at $x = x_\mathrm{p}$.

where $k_i = \frac{1}{2}L\left(EI\beta_i^4 + T\beta_i^2\right)$ and $r_i = \frac{1}{2}L\gamma(\beta_i)$ are the elastic and damping constants of the mode, respectively. The term $m = \frac{1}{2}\rho AL$ is the modal mass, which is the same for all modes. Under the assumption of each mode being *under-damped* (i.e. $r_i < 2\sqrt{k_i m}$), the modal frequencies are $\omega_i = \sqrt{k_i/m - \alpha_i^2}$, where (in accordance with (7))

$$\alpha_i = r_i/(2m) = \sigma_0 + \sigma_1\beta_i + \sigma_3\beta_i^3 \tag{12}$$

are the modal decay rates. The modal expansion of the termination force is

$$F_\mathrm{T}(t) = \sum_{i=1}^{M} \underbrace{\Big[-Tv_i'(L) + EIv_i'''(L)\Big]}_{w_i}\bar{y}_i(t), \tag{13}$$

where $v_i'(x)$ and $v_i'''(x)$ denote the first and third spatial derivative of $v_i(x)$, respectively.

The dynamics of each of the modes can be simulated in discrete time by discretising (11), for example using the impulse-invariant method [19], which exactly preserves the modal parameters ω_i, α_i. Denoting the transfer functions of the resulting digital model oscillators with $H_i(z)$, a modal synthesis structure that implements equation (10) then takes the form as illustrated in Figure 7; this modal sound synthesis engine structure is essentially the same as those proposed in various earlier studies (see, e.g. [3, 20]).

Two instances of this processing structure are created in order to simulate vibrations in two polarisations; this allows emulating beating effects due to a slight difference in effective length between the $y-$ and $z-$polarisations.

3.3 Real-Time Parameter Control

An early real-time prototype has been implemented in Max MSP [3] using the *resonators~* [4] object for the realisation of 1024 modal oscillators. The *resonators~* parameters are calculated with a dedicated external that translates MIDI controlled string parameters into modal parameters. This external utlises a frequency envelope function in order to avoid rendering aliased modes or clicks when varying parameters that affect the mode frequencies, ensuring that modes smoothly fade out when nearing the Nyquist frequency and fading in when the mode frequency falls below Nyquist.

[3] https://cycling74.com/products/max/

[4] http://cnmat.berkeley.edu/files/maxdl/archive/CNMAT_Externals-MacOSX-1.0-78-gd490ddd.tgz

4. PRELIMINARY EXPERIMENTS

In order to get a glimpse of what a virtual-acoustic string instrument of the proposed design might sound like, various explorative experiments were carried out. Piezo signals were recorded during a session in which a player applied forces to the string using various exciters, including a finger, plectrum, and a bow. These signals were processed off-line using the calibration filters in order to obtain estimations of the applied force signals, which were in turn fed to the modal resonator synthesis engine described in Section 3. For comparison, the modal resonator was also driven directly with the piezo signals, which yields sounds having the 'nasal' timbre typically associated with piezoresistive disks. This effect is significantly reduced by the calibration filters. Sound examples can be found on the accompanying webpage [5] . Further off-line exploration focused on using 'out of range' geometrical string parameters (length, cross-section), which allows exposing inherent string properties such as stiffness on a different time scale.

5. CONCLUSIONS

Physical models have been developed and implemented in real-time for several decades now. A rare example of turning a physical model into an exciting new virtual-acoustic instrument is the Kalichord [21], which departs from the configuration discussed in this paper in that it incorporates physical controller features of kalimbas and accordions in its design. The off-line results presented here are intended to give an initial impression of the wider possibilities of virtual-acoustic string instruments if specific attention is given to controller design that attempts to capture the interface dynamics in the form of an estimated excitation signal.

Some promising initial results are obtained, but several improvements are needed to more fully achieve the intended aims. Firstly, the signal conditioning needs to be improved in order to meet the signal-to-noise ratio requirements for this type of application. Secondly, in order to develop the potential of the approach more fully, the design needs to be targeted to more specific instruments, probably using extended models with well-tuned parameters. Finally, the next versions of the string controller will have to be more robust end ergonomic for application in performance.

A further consideration for future exploration is of a more abstract nature. The motivation behind playing a virtual rather a real resonator stems form the fully parameterisated nature of the virtual, i.e. one is free to change any of the physical parameters, thus having an instrument that houses a broad family of a certain type rather than one fixed instantiation. As discussed in [22], this concept can be extended to on-line variation of physical parameters that are not normally accessible in real-world instrument. Thus, once the virtual-acoustic instrument is functioning well in the sense of emulating both the acoustic and interface dynamics, an adventurous next step would be to explore extended control by real-time adjustment of a wider range of the available physical parameters.

[5] www.socasites.qub.ac.uk/mvanwalstijn/smc16/

6. REFERENCES

[1] V. Välimäki, J. Pakarinen, C. Erkut, and M. Karjalainen, "Discrete-time modelling of musical instruments," *Reports on Progress in Physics*, vol. 69, no. 1, pp. 1–78, Jan. 2006.

[2] J. O. Smith III, *Physical Audio Signal Processing: for Virtual Musical Instruments and Digital Audio Effects*. W3K Publishing, 2010.

[3] L. Trautmann and R. Rabenstein, *Digital Sound Synthesis by Physical Modeling Using the Functional Transformation Method*. Kluwer Academic/Plenum Publishers, New York, 2003.

[4] S. Bilbao, *Numerical Sound Synthesis: Finite Difference Schemes and Simulation in Musical Acoustics*. Chichester, UK: John Wiley and Sons, 2009.

[5] J. Bowers and S. O. Hellström, "Simple Interfaces to Complex Sound in Improvised Music," in *CHI '00 Extended Abstracts on Human Factors in Computing Systems*, ser. CHI EA '00. New York, NY, USA: ACM, 2000, pp. 125–126.

[6] A. Hunt and R. Kirk, "Mapping strategies for musical performance," *Trends in Gestural Control of Music*, vol. 21, pp. 231–258, 2000.

[7] T. Mudd, S. Holland, P. Mulholland, and N. Dalton, "Investigating the effects of introducing nonlinear dynamical processes into digital musical interfaces," in *Proceedings of Sound and Music Computing Conference, 2015*. Sound and Music Computing Network, Jul. 2015.

[8] M. M. Wanderley, "Gestural control of music," in *International Workshop Human Supervision and Control in Engineering and Music*, 2001, pp. 632–644.

[9] D. Menzies, "Composing instrument control dynamics," *Organised Sound*, vol. 7, no. 3, pp. 255–266, Dec. 2002.

[10] C. Desvages and S. Bilbao, "Two-Polarisation Finite Difference Model of Bowed Strings with Nonlinear Contact and Friction Forces," in *Proc. of the 18th Int. Conference on Digital Audio Effects*, Trondheim, Norway, Dec. 2015.

[11] S. Mann, "Natural Interfaces for Musical Expression: Physiphones and a physics-based organology," in *Proceedings of the 7th international conference on New interfaces for musical expression*. ACM, 2007, pp. 118–123.

[12] E. Hornecker, "A Design Theme for Tangible Interaction: Embodied Facilitation." in *ECSCW*, vol. 5. Springer, 2005, pp. 23–43.

[13] S. O'Modhrain and G. Essl, "PebbleBox and CrumbleBag: tactile interfaces for granular synthesis," in *Proceedings of the 2004 conference on New interfaces for musical expression*. National University of Singapore, 2004, pp. 74–79.

[14] P. Bennett, N. Ward, S. O'Modhrain, and P. Rebelo, "DAMPER: a platform for effortful interface development," in *Proceedings of the 7th international conference on New interfaces for musical expression*. ACM, 2007, pp. 273–276.

[15] E. Berdahl and J. O. Smith III, "A Tangible Virtual Vibrating String," in *Proceedings of the 2008 Conference on New Interfaces for Musical Expression (NIME08)*, 2008.

[16] C. Zheng and Y. Knobloch, "A Discussion of the History of the Gu zheng," *Asian Music*, vol. 14, no. 2, p. 1, 1983.

[17] S. Haykin, *Adaptive Filter Theory*, 5th ed. Upper Saddle River, N.J.: Pearson Education, Jun. 2013.

[18] M. Karjalainen and J. Smith, "Body Modeling Techniques for String Instrument Synthesis," in *CCRMA Papers Presented at the 1996 International Computer Music Conference Hong Kong*. Hong Kong, China: CCRMA, 1996, pp. 35–42.

[19] J. G. Proakis and D. G. Manolakis, "IIR Filter Design by Impulse Invariance," in *Digital Signal Processing*, 2nd ed. New York, USA: Macmillan Publishing Company, 1992, pp. 623–627.

[20] F. Avanzini, R. Marogna, and B. Bank, "Efficient synthesis of tension modulation in strings and membranes based on energy estimation," *The Journal of the Acoustical Society of America*, vol. 131, no. 1, pp. 897–906, 2012.

[21] D. Schlessinger and J. O. Smith, "The Kalichord: A Physically Modeled Electro-Acoustic Plucked String Instrument." in *NIME*. Citeseer, 2009, pp. 98–101.

[22] S. Orr and M. van Walstijn, "Modal representation of the resonant body within a finite difference framework for simulation of string instruments," in *12th International Conference on Digital Audio Effects, Como, Italy*, 2009.

DETECTION THRESHOLDS IN AUDIO-VISUAL REDIRECTED WALKING

Florian Meyer
Hamburg University of Technology
fl.meyer@tuhh.de

Malte Nogalski
Hamburg University of Applied Sciences
malte.nogalski@haw-hamburg.de

Wolfgang Fohl
Hamburg University of Applied Sciences
wolfgang.fohl@haw-hamburg.de

ABSTRACT

Redirected walking is a technique that enables users to explore a walkable virtual environment that is larger than the extent of the available physical space by manipulating the users' movements.

For the proper application of this technique, it is necessary to determine the detection thresholds for the applied manipulations. In this paper an experiment to measure the detection levels of redirected walking manipulations in an audio-visual virtual environment is described and the results are presented and compared to previous results of a purely acoustically controlled redirected walking experiment.

1. INTRODUCTION

In the design of interactive media environments, it is often desirable to create a virtual space that is larger than the physical space of the reproduction room. For the creation of *walkable* virtual environments, the technique of *redirected walking (RDW)* can be applied to extend the virtual area walkable for the user.

The basic idea is to apply *gains* to the user's turns and walking paths in order to manipulate the physical paths in a way that the user stays within the borders of the physical environment. For a proper immersion into the virtual environment, the applied gains must remain below the user's detection threshold. Our paper reports experiments to determine the detection thresholds of *curvature* and *rotational gains* in an audio-visual virtual environment.

It is generally accepted that vision dominates audition in 3D-orientation of persons (Goldstein [1] cited by Serafin [2]). The open question is, what the consequences for redirected walking (RDW) detection thresholds and thus the possibility to manipulate users' movements are. According to Lackner [3], cited by Razzaque et al. [4], a consistent set of various sensual cues will *increase* the detection threshold of manipulations, i.e., a RDW manipulation is less likely to be detected by the user, and thus larger gains may be applied.

After having previously conducted a purely auditive RDW experiment [5], we now executed the same experiment with an added visual component in order to check the above mentioned hypothesis. This audio-visual RDW experiment and the comparison to the audio-only experiment is the subject of this paper.

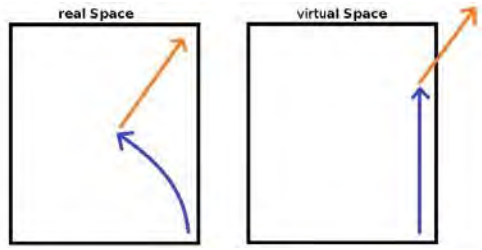


Figure 1: Walking paths in real space and virtual space.

In the following sections, first an introduction to the basic concepts of redirected walking and the current state of research is given. After that, the test procedure and the setup and architecture of the experiment environment is described. Then the results of our experiments are presented and compared to previous results with pure acoustically controlled RDW. Finally, a discussion of the results and an outlook on future work is given.

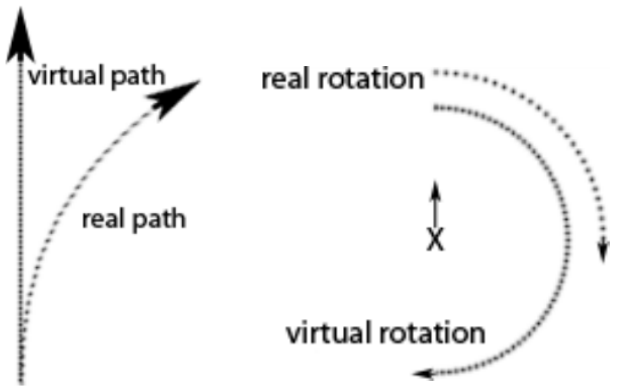
2. FUNDAMENTALS AND RELATED WORK

This section will give an introduction to the basic concepts of RDW. Various approaches to apply gains to manipulate users' movements are reviewed, and the reported thresholds for the identification of these manipulations are summarized for both visually and non-visually guided RDW.

2.1 General Redirected Walking (RDW)

Humans rely primarily on vestibular, visual and auditory cues for balance and orientation [6], and these cues are also used to distinguish between self-motion (the user moves) and external-motion (the objects around the user, respectively the immersive virtual environment (IVE), move). Under certain circumstances external-motion may be perceived as self-motion, and a consistency of multiple orientation cues may increase that chance [3]. By carefully manipulating the virtual environment (VE), RDW evokes a perceived self-motion of the user, and such provokes an automatic and unconscious self-motion to compensate for the manipulation.

RDW algorithms usually try to steer the users towards the center or the farthest wall of the physical tracking area [7],



(a) The curvature gain bends a real path into a distorted virtual path. The user unknowingly walks on a curved path.

(b) The rotation gain scales a rotation with the effect that the virtual rotation is greater or smaller than the real rotation.

Figure 2: The curvature gain bends a path and the rotation gain scales a rotation.

while the user is unaware of the steering process and can roam freely. RDW aims at providing the exploration of an infinite IVE within a confined tracking area [8].

2.2 Gains to Manipulate the Users' Movements

While the tracking system constantly provides up-to-date data for the user's physical world position and orientation defined as $P_{physical}$ and $R_{physical}$, the translation is defined by

$$T_{physical} = P_{cur} - P_{pre} \tag{1}$$

where P_{cur} is the current physical position and P_{pre} the previous/last considered physical position. The physical rotation is defined by

$$R_{physical} = R_{cur} - R_{pre} \tag{2}$$

By gains a discrepancy between physical and the virtual movements $T_{virtual}$ and $R_{virtual}$ can be dynamically applied.

A curvature gain stimulates users to unknowingly walk an arc in the tracking area while walking on a straight line in the VE even when they do not intentionally rotate (see figure 2a). A curvature gain g_C is defined by the radius r of the complete circle defined by the curve:

$$g_C = \frac{1}{r} \tag{3}$$

The particular rotational manipulation R_Δ is then calculated by multiplying the physical translation with the curvature gain value:

$$R_\Delta = T_{physical} \cdot g_C \tag{4}$$

R_Δ is then applied to the IVE, but perceived as self-motion by the user.

Rotation gains g_R scale a user's rotation to in- or decrease the amount of a user's virtual rotation $R_{virtual}$ in respect to $R_{physical}$ as illustrated in figure 2b, and are preferably calculated with the rotation of the user's head:

$$g_R = \frac{R_{physical} - R_{virtual}}{R_{physical}} \tag{5}$$

The particular rotational manipulation R_Δ is then calculated by multiplying the physical rotation with the rotation gain value:

$$R_\Delta = R_{physical} \cdot g_R \tag{6}$$

Figure 2b illustrates a rotation gain with a value g_R(-0.5), which up-scales a physical rotation of 90° to a virtual rotation of 180°.

2.3 Experiments for Detecting Thresholds

In March 2008 Steinicke at al. published results of a pilot study [9] within a tracking range of 10m x 7m x 2.5m, in which they identified the following thresholds for visual RDW: Rotations could be compressed or gained up to 30%, distances could be downscaled to 15% and up-scaled to 45%, users could be redirected to unknowingly walk on a circle with a radius as small as 3.3m, and objects and the VE could be down-scaled to 38% and up-scaled to 45%.

The results of different experiments differ greatly though. Other experiments identified thresholds for manipulated rotations at 49% for up-scaling and 20% for down-scaling, as well as a radius for a curved path of 22 meters [10], or 68% for up-scaling, and 10% for down-scaling rotations [11]. The differences in detection thresholds probably correlate with the attention that the test subjects actively pay to the manipulations [10] or other context specific parameters.

2.4 Non-Visual Redirected Walking by Acoustic Stimuli

While a lot of research has been committed to RDW during the last decade, almost all contributions are based upon the visualization of the VE for primary stimuli. Some authors state that the acoustic factor helps users to adjust to the virtual world and that RDW works best, when multiple cues, such as vestibular, visual and auditory, are consistent with each other. This should help the user to perceive external-motion as self-motion [3, 4], and a fully spatialized 3D sound model should be an important component of an IVE for RDW [4]. Even though, the auditory aspect had been paid little attention so far [2].

To the authors' knowledge, Serafin et al. are the only ones who really concentrated on the auditory component of RDW techniques. They conducted two different experiments to determine thresholds for acoustic based RDW techniques [2]. To that goal, they adapted two of the experiments conducted in [10, 11], to be used exclusively with auditory cues. Their experimental setup consisted of a surround system with 16 MB5A Dynaudio speakers in a circular array with a diameter of 7.1 meters and subjects wore a deactivated head mounted display (HMD) to block out their vision. The only audible stimulus in both experiments was the sound of an alarm clock. The sound was delivered

through the speaker array by the technique of vector base amplitude panning (VBAP). In such a setup, VBAP allows the placement of sounds within the circular array of speakers on a plane parallel to the ground level [12].

The first experiment tested the ability to detect rotation gains during rotations on the spot. The second experiment tested the detection of curvature gains while walking on a virtually straight line from one edge of the circular speaker array to a point roughly on the opposite side.

During the first experiment the subjects were asked to turn on the spot towards the sound of the alarm clock. While they were turning, a rotation gain would rotate the alarm clock around the subjects. A rotation gain > 0 would rotate the alarm clock in the same direction the subject is turning, and therefore making it necessary to turn further, to finally face the alarm clock. A rotation gain < 0 would have the opposite effect and result in a smaller physical rotation. When they perceived the sound as in front of them, they were asked whether they perceived the virtual rotation as larger (rotation gain < 0) or smaller (rotation gain > 0) than the physical rotation. The virtual rotation is perceived through auditory cues by locating the position of the sound source, while the physical rotation mainly by the vestibular and proprioception system. During the 22 subsequent trials per test subject, 11 different rotation gains were applied. Each gain was applied twice during the course of an experiment. For the evaluation Serafin et al. also oriented themselves at [10]. Serafin et al. also chose an outbalance of 75% to 25% of the given answers as the detection threshold and these thresholds were reached at gains of 0.82 for greater and 1.2 for smaller responses. This led them to the conclusion, that users can not reliably distinguish between a $90°$ physical rotation and a virtual rotation between $75°$ and $109°$. So users can be turned 20% more or 18% less than the perceived virtual rotation. This range is smaller than a corresponding experiment reported in [10], which can be attributed to the fact, that "[...] vision generally is considered superior to audition when it comes to the estimation of spatial location of objects." Goldstein [1] cited by Serafin et al. [2].

During the second experiment users were asked to walk on a straight line towards the alarm clock. During their movements 10 different curvature gains were applied (each one twice), which led them on an arced physical path and users were asked whether and at which threshold they noticed the direction of the bent path reliably. For this experiment the curvature gain value was defined as the degree the scene rotated after the test subjects walked the whole path of 5 meters. During this experiment the point of subjective equality (PSE) was determined at a curvature gain of -5. The detection thresholds of 75% were reached at gains of -25 and 10 [1] [2]. 25 is roughly equivalent to a circle with a radius of 11.45 meters.

[1] S. Serafin confirmed in personal correspondence that a mistake slipped into the textual representation of the results. Instead of +30, +10 is correct (as the corresponding plot of the paper illustrates)

2.5 Audio-Visual Rotational Gains

The interdependence of acoustical and visual stimuli has recently been investigated by Nilsson et al. [13] for detection thresholds of rotational gains. They conducted experiments *without audio*, with *static audio* (i.e., visual and acoustical targets are primarily lined up, then the gain is only applied to the visual environment), and *moving audio* (i.e., the audio source position moves consistently with the visual scene). The experiments resulted in no significant influence of audition on the detection rates.

2.6 Cyber Sickness

Since RDW manipulates the visual and/or auditory cues willfully and the discrepancy between cues of different senses can lead to different kinds of sicknesses, the consideration and measurement of cyber sickness is part of most experiments regarding RDW.

3. METHODOLOGY AND EXPERIMENTAL SETUP

3.1 Experiment Design

Our experiment for the detection of audio-visual redirected walking was designed ¡¡¡¡¡¡¡ HEAD in close correspondence with our previous audio-only experiment. ======= in close correspondence with our previous audio-only experiment [14]. ¿¿¿¿¿¿¿ release/1.0

To reduce problems with participants suffering from simulator sickness, the range of gains, and such the number of tests were reduced after the first 8 experiment runs.

In the modified experiment design, a complete experiment run for each participant consisted of 44 curvature and 28 rotational gain tests. Ranges of gain values were reduced from $[-1 \dots 1]$ to $[-0.6 \dots 0.6]$ for curvature gains, and from $[-0.6 \dots 0.6]$ to $[-0.4 \dots 0.4]$ for rotational gains. The selection of the test sequence was performed at random by the experiment control software.

Each curvature gain task consisted of the following steps:

1. Participant walks to a corner of the test area (see Fig. 3),
2. participant turns in the direction of one of the adjacent corners,
3. the audio-visual target appears,
4. participant walks towards the audio-visual walking target, while a curvature gain is applied.

These were the steps for a rotational gain test:

1. Participant walks to the center of the room (see Fig. 3),
2. participant turns towards one of the sides of the test area,
3. the audio-visual target appears at $180°$, i.e., directly behind the participant,
4. participant turns towards the audio-visual target, while a rotational gain is applied.

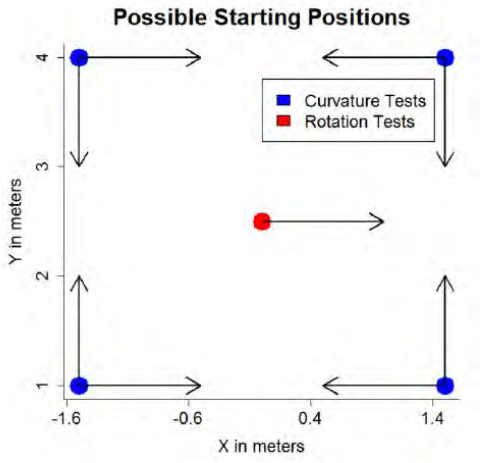


Figure 3: Starting positions for RDW experiments.

Detailed explanations on the design of the test procedure can be found in [14].

To perform the tasks, three simple instructions were given to the participants:

Figure 4: Left: the visual target for walking, right: the visual target for rotation.

1. If you hear or see the *walking target* (see Fig. 4, left), walk towards it, until it vanishes.
2. If you hear or see the *rotation target* (see Fig. 4, right), turn into the direction of the target, until it vanishes.
3. Give feedback ("left" or "right") about the perceived manipulations.

The purpose of the experiment, determination of RDW detection thresholds, was explained to the participants in advance, but neither the participants nor the experiment operator knew the detailed sequence of tests, since they were randomly selected by the test control software. Tests were carried out as *two-alternative forced-choice* tests (2AFC): after each test, the participants only had the choice between the responses "right", indicating a manipulation towards the right-hand side, or "left" for a manipulation to the opposite direction. The answer "no manipulation", or no answer at all was not allowed. In this case the participants had to guess. As a consequence, with no gain applied, the reported left and right manipulations are expected to be equal.

Before and after the experiment, the participants filled a simulator sickness questionnaire according to [15].

Figure 5: The visual virtual environment. The oasis (1), camp (2), pyramid (3), and village (4) are orientation marks. The red circle is the area of user movement.

3.2 System Setup

3.2.1 The Visual Component

The visual component of the virtual environment was designed with Unity [2] for the Oculus Rift DK2 [3]. The scenery is given in Fig. 5. It shows an desert-like area surrounded by distant orientation marks: an oasis, a camp, a pyramid and a village. The red circle in the center indicates the area accessible to the user. It has an diameter of $11\,\mathrm{m}$. To adapt the Oculus Rift to our walking experiments, the Oculus tracking system was substituted by the 3D-tracking system of our lab. Thus the available physical space was a rectangle of approx. $3\times4\,\mathrm{m}$. The computer controlling the Oculus Rift was carried in a backpack by the user as shown in Fig. 6.

3.2.2 The Acoustical Component

The acoustical component serves two purposes: to generate the sounds for the direction of the participants, and to create background noises that provide acoustical landmarks for orientation and for masking of real-world background noises in the lab.

The core part of the acoustic component is a WFS system [16] to create the desired spatial sounds. A very comprehensible overview of the principles of WFS had been given by Spors and Zotter in a tutorial held at the 138^{th} AES convention [17], a thorough analysis is given in the book of Ahrens [18].

The rendering software *sWonder* [4] has been modified by our team to provide proper spatial rendering of focused sources regardless of the participant position [19]. Sounds are played back by a DAW software running on the control computer (see Fig. 7).

The background noises were sounds of flamingos, camels, a campfire, oriental music and wind. The background sounds

[2] http://unity3d.com/
[3] https://www.oculus.com/en-us/rift/
[4] https://github.com/sensestage/swonder

Figure 6: A fully equipped test person with tracking system target (1) and Oculus Rift (2) connected via cable (3) to the control laptop (4).

were rendered as plane waves, arriving from the directions of the visual landmarks.

The controlling sounds for the participants were the sparkling sound of a fountain as *acoustic walking target*, and the sound of a barking dog as *acoustic rotation target*.

3.2.3 System Architecture

The overall system architecture is given in Fig. 7. The system is built with these components:

1. An IR-based tracking system that broadcasts the participant's position via WLAN,
2. the Oculus Rift and its controlling laptop PC carried by the participants in a backpack,
3. a laptop PC controlling the movements of the virtual sound sources,
4. a control computer running these programs:
   - DAW software for sound playback,
   - OSC gateway to the WFS system,
   - communication gateway for the Oculus Rift control PC,
5. the WFS system consisting of a controller PC and two rendering nodes for 208 speaker channels with a spacing of 10 cm forming an rectangle of roughly 5×6 m.

The connections between the laptop controlling the acoustic component (3), the control computer (4), and to the WFS system (5) are wired LAN connections, the other network connections are established via WLAN.

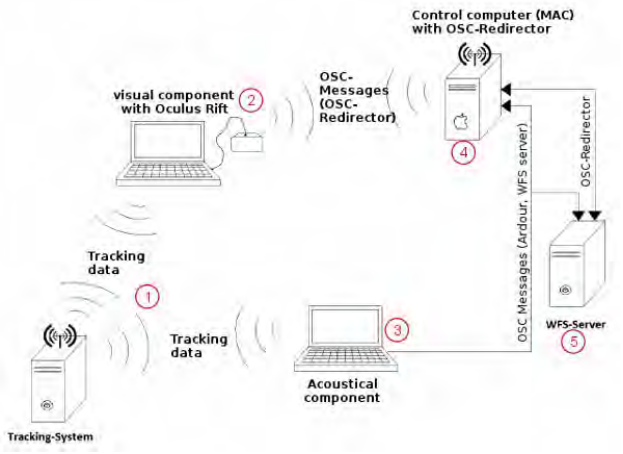


Figure 7: Distributed architecture of the experiment control system. Numbers 1 to 5 correspond to the enumeration given in the text

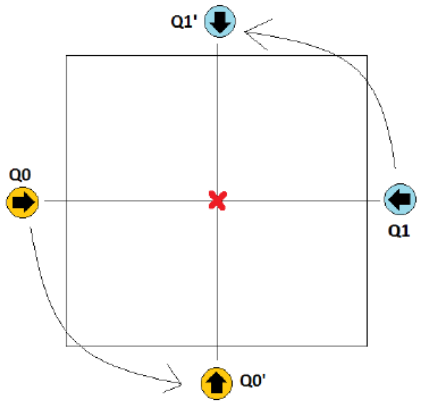


Figure 8: Rotating the acoustical environment.

3.3 Realization of Rotational and Curvature Gains

From our previous acoustical RDW experiments, there existed a fully functional software for the acoustical component to control the test sequences and apply the required gains. Gains are applied by rotating the virtual sound sources synchronously around the participant (see Fig. 8). This is being done by sending appropriate OSC messages to the WFS server via LAN (see Fig. 7). The WFS server offers a data stream with source positions. This stream is subscribed by the visual component to synchronize the visual with the acoustical environment.

4. RESULTS AND DISCUSSION

Our study was carried out with 20 participants, 8 female, and 12 male, most of them students of computer science. Each experiment lasted approximately one hour, with an uninterrupted exposure to the virtual environment of 20 – 35 minutes.

As stated in the previous section, our primary intention was to perform the same series of experiments that were earlier performed with purely acoustically controlled RDW, but it turned out that the participants suffered from severe

symptoms of simulator sickness. Therefore the applied gains were limited to smaller values.

The results are summarized in Fig. 9 for rotational gains, and in Fig. 10 for curvature gains. For the curvature gains, positive and negative gains have been pooled, as there was no significant bias in one of the directions.

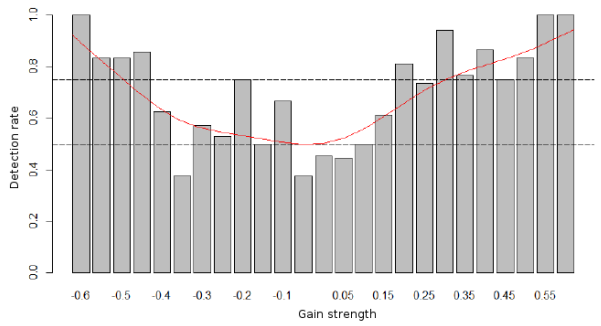


Figure 9: Results of the rotational gain experiments.

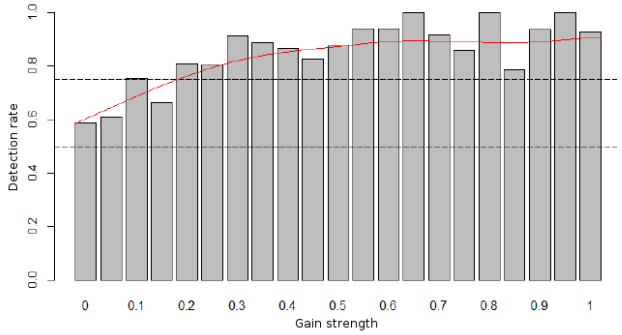


Figure 10: Results of the curvature gain experiments. Positive and negative curvature tests have been combined.

Since the test was designed as a two-alternative forced-choice test, a detection rate of $50\,\%$ indicates the *point of subjective equality* (PSE), because the participants had to give a feedback, either "left" or "right", after each run. A detection rate of $50\,\%$ would then be the result of simply *guessing*.

From the cubic spline drawn in Figs. 10 and 9, the gains for $62.5\,\%$ and $75\,\%$ detection rate can be estimated. The results are summarized in Table 1. The percentual angular ranges for undetected rotations are calculated using Eq. 5. With $R \equiv R_{physical}$, $R - R_\Delta \equiv R_{virtual}$, and $R_{virtual} = 180°$, this equation becomes:

$$R_\Delta = g_R \cdot (180° + R_\Delta) \tag{7}$$

$$R_\Delta = \frac{180°}{1 - g_R} \tag{8}$$

$$\frac{R_\Delta}{R} = \frac{180°}{R \cdot (1 - g_R)} \tag{9}$$

Our results can be compared with similar experiments, to get insight in the role of auditory and visual cues in

Type	Det. rate	Gain	Undetected
Rotation	$75\,\%$	$-0.463 \ldots 0.265$	$-57° \leq R_\Delta \leq 64°$
Rotation	$62.5\,\%$	$-0.366 \ldots 0.150$	$-48° \leq R_\Delta \leq 32°$
Curvature	$75\,\%$	$0.166\,\mathrm{m}^{-1}$	$r \leq 6.024\,\mathrm{m}$
Curvature	$62.5\,\%$	$0.031\,\mathrm{m}^{-1}$	$r \leq 32.3\,\mathrm{m}$

Table 1: RDW detection threshold summary.

RDW. Table 2 compares our data with the results of a visual RDW experiment reported by Steinicke et al. [10], and acoustical RDW experiments by Serafin et al. [2], and by our team [5]. The reported data are transformed to a uniform format and rounded to two significant digits for comparability: Rotational thresholds are given as relative angular manipulations in $\%$ according to Eq. 9, curvature thresholds are given as radii of circles that are perceived as straight paths. All listed data are the $75\,\%$ thresholds in 2AFC-experiments.

Author	Rotation	Curvature
This paper (A + V)	$-32\,\%$ ... $36\,\%$	$6.0\,\mathrm{m}$
Steinicke (V)	$-20\,\%$ ... $49\,\%$	$22\,\mathrm{m}$
Serafin (A)	$-18\,\%$ ... $20\,\%$	$16\,\mathrm{m}$
Our team (A)	$\leq -38\,\%$ ... $18\,\%$	$3.6\,\mathrm{m}$

Table 2: Comparison of RDW detection thresholds with results of other authors. A = acoustic, V = visual RDW control.

When comparing the first and last rows of Table 2, the data tends to contradict the starting hypothesis. It was to be expected that the detection thresholds for the audio-visual experiment (row 1) are higher than the thresholds for the audio-only experiment (row 4), but this is only true for *positive rotation gain* thresholds. For *negative* rotation gains, as well as for *curvature gains*, the gain thresholds are higher in the audio-only experiment.

4.1 Simulator Sickness

Many participants experienced simulator sickness symptoms, especially at higher gain values. In the first 8 runs, some tests even had to be interrupted, when participants complained about nausea. As a consequence, the tests with high gain values were skipped in the subsequent experiments. The average pre-SSQ score was 4.68, the average post-SSQ score was 32.25. The nausea score showed the highest increase, from 2.39 to 42.39. For comparison: In our acoustic RDW experiment, the pre-score was 2.33, and the post-score was 14.0.

A higher SSQ score for the audio-visual experiments had to be expected, since even without manipulations, some people feel uncomfortable watching the virtual scenes of an Oculus Rift.

5. SUMMARY AND OUTLOOK

Experiments to determine detection thresholds for rotational and curvature gains for audio-visual RDW have been car-

ried out. An overview of our results compared to other publications is presented in Table 2. Besides some great variances in the data, a close look shows a slight trend for higher positive rotational gain thresholds when visual cues are present, and for higher curvature gain thresholds with acoustical cues only. From the great variances in experiments of similar type however, it has to be concluded that there are many more factors to be considered, as for instance prior knowledge of the participants, or the consistency of acoustic and visual cues. To shed some light on the latter point, experiments will have to be executed with diverging acoustic and visual gains. As a consequence of the results of Nilsson et al. [13], who did not detect significant differences in their setup (see section 2.5), experiments with more drastic diversions in the acoustical and visual stimuli will be conducted.

Finally, it may be stated that audio-visual redirected walking does not only provide a method to create large virtual spaces in small physical rooms, but it also enables artists to create audio-visual environments that play with the interdependencies of sight, sound, and motion.

6. REFERENCES

[1] E. B. Goldstein, *Sensation and Perception*. Boston, MA: Cengage Learning, 2010.

[2] S. Serafin, N. Nilsson, E. Sikstrom, A. De Goetzen, and R. Nordahl, "Estimation of detection thresholds for acoustic based redirected walking techniques," in *Virtual Reality (VR), 2013 IEEE*, March 2013, pp. 161–162.

[3] J. Lackner, "Induction of illusory self-rotation and nystagmus by a rotating sound-field," *Aviation, space, and environmental medicine*, vol. 48, no. 2, pp. 129–131, February 1977.

[4] S. Razzaque, Z. Kohn, and M. C. Whitton, "Redirected walking," University of North Carolina at Chapel Hill, Chapel Hill, NC, USA, Tech. Rep., 2001.

[5] M. Nogalski and W. Fohl, "Acoustic redirected walking with auditory cues by means of wave field synthesis," in *Proc. 23rd IEEE Conf. on Virtual Reality*. IEEE, March 2016.

[6] J. Dichgans and T. Brandt, "Visual-vestibular interaction: Effects on self-motion perception and postural control," in *Perception*, ser. Handbook of Sensory Physiology, R. Held, H. Leibowitz, and H.-L. Teuber, Eds. Springer Berlin Heidelberg, 1978, vol. 8, pp. 755–804.

[7] E. Hodgson and E. Bachmann, "Comparing four approaches to generalized redirected walking: Simulation and live user data. visualization and computer graphics," *IEEE Transactions on Computer Graphics*, vol. 19, no. 4, pp. 634 – 643, 2013.

[8] P. Lubos, G. Bruder, , and F. Steinicke, "Safe-&-round: bringing redirected walking to small virtual reality laboratories," in *Proc. 2nd ACM symposium of spatial user interaction*. ACM, 2014, pp. 154 – 154.

[9] F. Steinicke, T. Ropinski, G. Bruder, K. Hinrichs, H. Frenz, and M. Lappe, "A universal virtual locomotion system: Supporting generic redirected walking and dynamic passive haptics within legacy 3d graphics applications," in *Virtual Reality Conference, 2008. VR '08. IEEE*, March 2008, pp. 291–292.

[10] F. Steinicke, G. Bruder, J. Jerald, H. Frenz, and M. Lappe, "Estimation of detection thresholds for redirected walking techniques," *Visualization and Computer Graphics, IEEE Transactions on*, vol. 16, no. 1, pp. 17–27, Jan 2010.

[11] ——, "Analyses of human sensitivity to redirected walking," in *Proceedings of the 2008 ACM Symposium on VRST*, ser. VRST '08. New York, NY, USA: ACM, October 2008, pp. 149–156.

[12] V. Pulkki, "Virtual sound source positioning using vector base amplitude panning," *J. Audio Eng. Soc*, vol. 45, no. 6, pp. 456–466, 1997. [Online]. Available: http://www.aes.org/e-lib/browse.cfm?elib=7853

[13] N. C. Nilsson, E. Suma, R. Nordahl, M. Bolas, and S. Serafin, "Estimation of detection thresholds for audiovisual rotation gains," in *Proc. 23rd IEEE Conf. on Virtual Reality*. IEEE, March 2016.

[14] M. Nogalski and W. Fohl, "Acoustically guided redirected walking in a WFS system: Design of an experiment to identify detection thresholds," in *Proc. 12th Sound and Music Computing Conf.* SMC, August 2015.

[15] R. S. Kennedy, N. E. Lane, K. S. Berbaum, and M. G. Lilienthal, "Simulator sickness questionnaire: An enhanced method for quantifying simulator sickness," *The International J. of Aviation Psychology*, vol. 3, no. 3, pp. 203–220, 1993. [Online]. Available: http://ci.nii.ac.jp/naid/30010213408/en/

[16] M. A. J. Baalman, "On wave field synthesis and electro-acoustic music, with a particular focus on the reproduction of arbitrarily shaped sound sources," PhD Thesis, TU Berlin, 2007.

[17] S. Spors and F. Zotter, "Tutorial: Foundations and Practical Aspects of Sound Field Synthesis," http://dx.doi.org/10.13140/RG.2.1.1025.5527, May 2015.

[18] J. Ahrens, *Analytic Methods of Sound Field Synthesis*. Springer Science & Business Media, 2012.

[19] W. Fohl and E. Wilk, "Enhancements to a wave field synthesis system to create an interactive immersive audio environment," in *Proc. 3rd Int. Conf. on Spatial Audio*. VDT, September 2015.

A FAUST BASED DRIVING SIMULATOR SOUND SYNTHESIS ENGINE

Romain Michon[1], Mishel Johns[2], Sile O'Modhrain[3], Nick Gang[1], Nikhil Gowda[2], David Sirkin[2], Chris Chafe[1], Matthew James Wright[1] and Wendy Ju[2]
[1]Center for Computer Research in Music and Acoustics (CCRMA), Stanford University
[2]Center for Design Research (CDR), Stanford University
[3]University of Michigan
rmichon@ccrma.stanford.edu

ABSTRACT

A driver's awareness while on the road is a critical factor in his or her ability to make decisions to avoid hazards, plan routes and maintain safe travel. Situational awareness is gleaned not only from visual observation of the environment, but also the audible cues the environment provides - police sirens, honking cars, and crosswalk beeps, for instance, alert the driver to events around them.

In our ongoing project on "investigating the influence of audible cues on driver situational awareness", we implemented a custom audio engine that synthesizes in real time the soundscape of our driving simulator and renders it in 3D. This paper describes the implementation of this system, evaluates it and suggests future improvements. We believe that it provides a good example of use of a technology developed by the computer music community outside of this field and that it demonstrates the potential of the use of driving simulators as a music performance venue.

1. INTRODUCTION

It is tempting to think that someday, when we have fully autonomous vehicles, we will be able to clamber into our cars and take a nap on the way to wherever we are headed as the vehicle takes over. And yet, consider how we behave today, when we have autonomous humans, such as taxi drivers, take us from point A to point B. Normally, we passengers feel the need to supervise drivers, at least somewhat, to make sure that they understand our intentions, that they are competent drivers, and that they are not lingering or adding unnecessary waypoints along the journey. Given the challenges associated with perceiving the roadway and associated obstacles, synthesizing many streams of data to develop a coherent model, and the inherently open-ended set of things that we encounter on the road every day, it is altogether likely that people will need to maintain some level of awareness over automation in cars for many years to come.

Situation awareness (SA) was defined by Endsley [1] as "the perception of the elements in the environment within a volume of time and space, the comprehension of their meaning, and the projection of their status in the near future." Such awareness of the surroundings and the situation is necessary for drivers of road vehicles, in order to maintain safe travel. Situational awareness is gleaned not only from visual observation of the environment, but also the audible cues the environment provides - police sirens, honking cars, and crosswalk beeps, for instance, alert the driver to events around them.

Prior work compares spatialized auditory feedback methods [2] and describes the use of spatialized audio for navigation cues [3].

In our ongoing project [1] at Stanford University's *Volkswagen Automotive Innovation Lab* (VAIL) on "investigating the influence of audible cues on driver situational awareness", we are trying to test the following hypotheses:

- Selective amplification/modification of explicit signals will increase the driver's situation awareness.
- Audible rendering of explicit events will affect the driver's spatial recall and sense-making of the surrounding events and the vehicle's actions.
- Selective amplification/modification of ambient signals will make the drive more enjoyable for the driver.
- Audible rendering of ambient context will affect spatial recall.
- Possible interaction effect where using sonification becomes distracting and makes the driver less effective in prescribed tasks.

The Stanford driving simulator uses the RTI [2] (*Realtime Technologies Inc.*) software to design and run the simulations in the frame of this study. While this environment provides a comprehensive set of tools to create virtual driving spaces, it has a quite limited synthesis engine that only renders the sound of the simulator car engine, the road and other cars passing by. Additionally, the sound of the simulation is spatialized only in 2D on a simple 5.1 audio system.

[1] This project is the fruit of a collaboration between Stanford University's *Center for Computer Research in Music and Acoustics* (CCRMA), *Center for Design Research* (CDR) and *Renault Innovation Silicon Valley*.

[2] http://www.simcreator.com/simulators.htm Accessed: 2016-07-11.

The simulator is equipped with an automated mode that was used to transport participants through a predefined course. During certain sections of the course, experimenters were required to take over control using a secondary steering wheel in a location not visible to participants. This is known as a "Wizard of Oz" control scenario, and it was used as a more felxible alternative to programming every autonomous movement in the simulation. To better present auditory cues from outside the car, the front windows were left cracked open about 10cm. The rear windows were fully closed.

The simulated course used for our study begins on a parking lot where a series of events happen sequentially around us: a child shouts amd runs right next to our car, a bicyclist rings a bell and crosses the road in front of us, a dog barks in reaction, and a garbage truck is heard in the distance (see Figure 1). Synthesizing and rendering all these sonic events required a much more versatile audio engine than the one at our disposal. We needed to be able to easily add any sound source to the simulation and to spatialize them in the three dimensional space of the simulator. We also wanted to improve the quality of the generated sounds by using physical models instead of samples when this was possible.

Figure 1. Screenshot of the beginning of the simulation used for our study.

While advanced audio engines for car simulators have already been implemented and described in previous work [4, 5], we decided to implement our own from scratch in order to custom tailor it to our needs and to make it open source.

In this paper, after providing a detailed description of the implementation of our custom audio engine, we evaluate it and provide suggestions for its improvement.

2. HARDWARE

2.1 Driving Simulator

Simulators are cost-effective ways to evaluate performance on tasks that are too dangerous or improbable in real life. They possess advantages over on-road vehicle testing in that the experimenter has better control over the vehicle surroundings and situations, and can provide for easier data collection [6].

The Stanford Driving Simulator is a high fidelity full-car automotive simulator installed by *Realtime Technologies, Inc*. The car is a 2010 *Toyota Avalon*, with connections to the onboard `CAN`, a set of digital and analog multipurpose inputs and outputs, and a `TCP/IP` interface. A mixed `WAGO` analog and digital I/O System is used by the host computer to send and receive signals from the car. Twenty-four digital I/O ports and 16 analog I/O ports can be used to connect the system to other devices. Steering feedback is provided by a control loading steering motor that returns torque to the wheel based on vehicle speed, suspension parameters and steering angles. Brakes use the OE hydraulic brake system with a vacuum pump replacing the vacuum normally supplied by the engine.

The visual system consists of a twenty-four foot diameter 260° cylindrical screen. Images from five high-resolution projectors are blended together form a seamless display. LCD inserts for the side mirrors and a rear projector reflected on a rear-view mirror complete the visual immersion in the simulated environment. The side mirror adjust buttons control the eye-points for these LCD inserts.

The Stanford Driving Simulator has an autonomous mode that replicates the experience of being in a self-driving vehicle. The system uses the simulations own traffic control algorithms to calculate the parameters to drive the car. The steering wheel moves on its own when the automated mode turns the car. The gas and brake pedals, however, do not move when the car is automated. The mode (autonomous or manual) of the car is displayed on an icon in the center of the instrument panel.

The pitching of the visuals due to acceleration and braking, as well as tight curves on the simulated roadway do tend to cause discomfort and simulator sickness in a small fraction of our participants. This is probably because of the immersive visual experience that makes the differences between the inputs to the visual and the vestibular systems more noticeable. The solutions that seem to help reduce nausea are to avoid tight turns in the course, enable the air conditioning system in the cabin with cold air blowing towards the driver, and to keep ginger candy and cold water available for participants. In this study, there were no sessions that needed to be halted due to simulator sickness.

Figure 2. The Stanford Driving Simulator.

2.2 Audio System

The sound system of the Stanford driving simulator consists of a simple sphere of speakers placed on three different levels:

- a ring of four speakers around the car,
- a ring of four speakers right above the roof of the car,

- a center speaker on the ceiling of the simulator room.

In addition to that, 4 speakers in the passenger's compartment of the car (two on each side of the front and back seats) and a large speaker mounted on the driver's seat to transduce road vibrations can be accessed. Speakers are controlled using a set of two audio interfaces [3] connected together using an `ADAT LightPipe` and plugged to a *Mac Mini*. The *Mac Mini* is connected to the simulator computer cluster via ethernet and runs the custom audio engine (see Figure 4). Having a dedicated computer to carry out this task made prototyping easier.

3. SOFTWARE

The sound synthesis engine of the Stanford driving simulator was implemented with the FAUST programming language [7] [4] and is compiled as a standalone C++ application with a `Qt` [5] user interface (see Figure 3). Each of its parameters can be controlled using `OSC` [8] [6] allowing the simulator software to interact in real time with the sound synthesis engine by sending raw `UDP` messages. Those messages are converted to `OSC` using a small program that we wrote in C++ and which runs independent of the other process (see Figure 4).

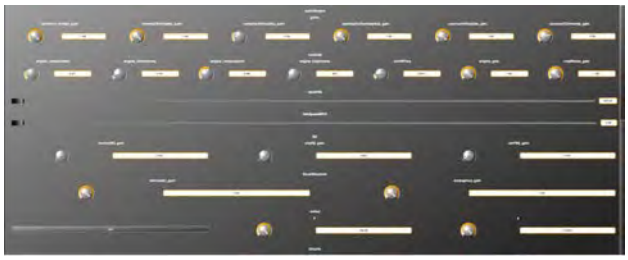


Figure 3. Screenshot of the FAUST generated user interface of the driving simulator sound synthesis engine.

All the generators of the sound synthesis engine render sounds in 3D and access the different speakers of the simulator independently. The signals sent to the speakers of the passenger's compartment are processed by fourth-order lowpass filters with a cut-off frequency of 500 Hz to reproduce the effect of the car's shell on interior sounds coming from the soundscape when windows are closed. Similarly, the signal of the driver's seat speaker is lowpass filtered at 90Hz.

A reverberator based on a FAUST implementation of *freeverb* [7] is used to model the effect of passing through closed structures like tunnels, etc. on the soundscape (see Figure 4).

The synthesized car engine sound is based on a physical model made out of an aperiodic pulsetrain generator fed through a series of filters and effects [9]. This kind of model was designed to simulate the sound of race cars for video games and was not well adapted to our use. Therefore we had to spend a fair amount of time tuning the model so that it would sound more like a small four cylinder engine of a *Toyota Avalon*. Since the *RPM* of the engine is controlled directly by the simulator software, we didn't have to implement features such as gear shifts, engine torque and resistance, etc.

The road noise is synthesized using a noise generator processed by a lowpass filter whose cut-off frequency and gain are adjusted in function of the speed of the car (see Figure 5). The faster the car, the greater the cut-off frequency and the gain.

An algorithm similar to the one presented in Figure 5 is used to synthesize the sound of other cars present in the simulation. The only difference comes from the fact that the cut-off frequency and the gain of the lowpass filter are calculated as a function of the distance between our car and the other cars.

The different sound sources of the simulation can be moved in the 3D space of the simulator by using a spatialization function (see Figure 6). The simulator software provides real-time cartesian coordinate of the different sound sources of the simulation relative to the position of our car. The audio engine converts those coordinates to a set of distances, angles (azimuths) and elevations to carry out *Vector Based Amplitude Panning* (VBAP) [10]. The first-derivative of distance drives a Doppler effect. Any sound source can be easily added to the audio engine by providing an audio sample that can be looped and sent to the spatialization function.

The different elements of the audio engine were condensed into a Faust library [8] in order to easily customize it and reuse it.

4. EVALUATION AND FUTURE DEVELOPMENTS

Faust's compactness enabled us to build our audio engine in less than 400 lines of code and the efficiency of the generated C++ code only requires 12% of the CPU of our *Mac Mini* when simultaneously calculating 28 sound sources, the car engine physical model, the sound of the road, a reverb and a series of filters (see Figure 4).

Neither is the "typical roaring sound" synthesized when the windows of the car are open and the car is moving. More transient sonic events could be added especially when driving through a city to improve the simulation's realism. Investigation of further details of the car engine sound model remains a top priority in our future work. We will eventually integrate computation of the entire sound system natively in the RTI simulator thereby eliminating the need for a separate sound computer.

5. CONCLUSIONS

We have built a versatile, fully customizable and efficient synthesis engine compatible with the RTI software used in the Stanford driving simulator. It is implemented in the FAUST programing language and it can be easily interfaced

[3] *Focusrite Scarlett 18i20.*
[4] `http://faust.grame.fr` Accessed: 2016-07-11.
[5] `http://qt.io` Accessed: 2016-07-11.
[6] *Open Sound Control.*
[7] `https://ccrma.stanford.edu/~jos/pasp/Freeverb.html` Accessed: 2016-07-11.
[8] `https://ccrma.stanford.edu/~rmichon/faustDrivingSimulator` Accessed: 2016-07-11.

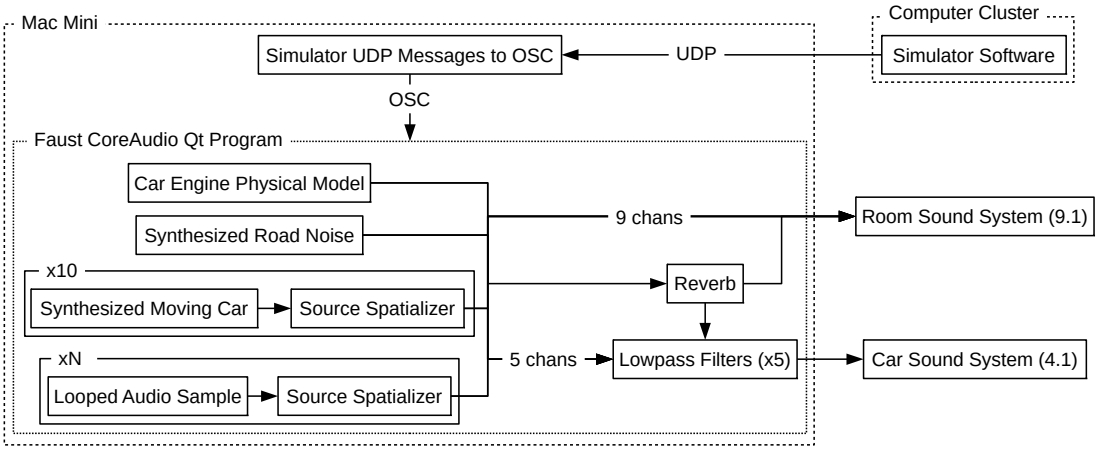


Figure 4. Overview of the implementation of the driving simulator synthesis engine.

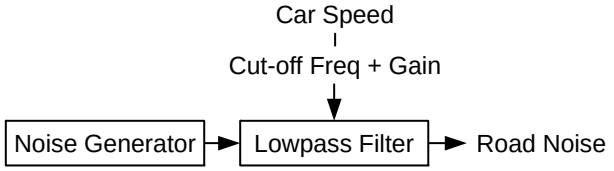


Figure 5. Overview of the implementation of the road noise generator.

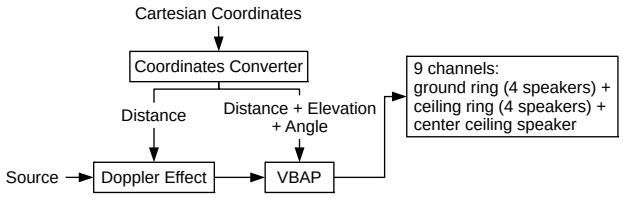


Figure 6. Overview of the implementation of the source spatialiazer.

with any software thanks to its `OSC` interface. We hope that it will help improve the driver's experience and provide an environment to design more advanced simulations and audio-related studies useful in the development of autonomous driving.

More generally, we believe that driving simulators provide a standardized complete environment to design interactive art installations. Investigating this unexploited potential should be greatly facilitated by our custom audio engine that provides an ideal prototyping environment to play with that kind of ideas.

Acknowledgments

We thank *Renault Innovation Silicon Valley* for funding this project as well *moForte Inc.* for allowing us to use their car engine physical model algorithm.

6. REFERENCES

[1] M. Endsley, "Toward a theory of situation awareness in dynamic systems," *Human Factors: The Journal of the Human Factors and Ergonomics Society*, vol. 37, no. 1, pp. 32–64, 1995.

[2] D. Beattie, L. Baillie, M. Halvey, and R. McCall, "What's around the corner? enhancing driver awareness in autonomous vehicles via in-vehicle spatial auditory displays," in *Proceedings of the 8th Nordic Conference on Human-Computer Interaction*, New York, NY, 2014, pp. 189–198.

[3] M. Cohen, O. N. N. Fernando, T. Nagai, and K. Shimizu, "Back-seat driver: Spatial sound for vehicular way-finding and situation awareness," in *Proceedings of the Japan-China Joint Workshop on Frontier of Computer Science and Technology (FCST'06)*, November 2006.

[4] M. Blommer and J. Greenberg, "Realistic 3d sound simulation in the virttex driving simulator," in *Proceedings of ASME DSC North America*, Dearborn, MI, October 2003.

[5] T. Funkhouser, N. Tsingos, and J.-M. Jot, "Survey of methods for modeling sound propagation in interactive virtual environment systems," Technical Repport, 2003.

[6] J. Gruening, J. Bernard, C. Clover, C. Clover, and K. Hoffmeister, "Driving simulation," *Vehicule Dynamic Simulation*, 1998.

[7] Y. Orlarey, D. Fober, and S. Letz, "An algebra for block diagram languages," in *Proceedings of the International Computer Music Conference (ICMC-02)*, Gothenburg, Sweden, 2002.

[8] M. Wright and A. Freed, "Open Sound Control: A new protocol for communicating with sound synthesizers," in *Proceedings of the International Computer Music Conference*, Thessaloniki, Greece, 1997.

[9] K. Cascone, D. T. Petkevich, G. P. Scandalis, T. S. Stilson, K. F. Taylor, and S. A. V. Duyne, "Apparatus and methods for synthesis of internal combustion engine vehicle sounds," US Patent US 6 959 094 B1, 2005.

[10] V. Pulkki, "Virtual sound source positioning using vector base amplitude panning," *Journal of the Audio Engineering Society*, vol. 45, no. 6, pp. 456–466, June 1997.

NUANCE: ADDING MULTI-TOUCH FORCE DETECTION TO THE IPAD

Romain Michon, Julius O. Smith III, Chris Chafe, Matthew Wright, and Ge Wang
Center for Computer Research in Music and Acoustics (CCRMA), Stanford University
{rmichon,jos,cc,matt,ge}@ccrma.stanford.edu

ABSTRACT

NUANCE is a new device adding multi-touch force detection to the *iPad* touch screen. It communicates with the *iPad* using the *audio jack* input. Force information is sent at an audio rate using analog amplitude modulation (AM). NUANCE provides a high level of sensitivity and responsiveness by only using analog components. It is very cheap to make.

NUANCE has been developed in the context of a larger project on augmenting mobile devices towards the creation of a form of hybrid lutherie where instruments are based on physical and virtual elements.

1. INTRODUCTION

Smartphones and tablets have been widely used as musical instruments and musical controllers during the last decade [1–6] (to only cite a few). In an era dominated by the controller/synthesizer paradigm where these two entities are often physically (or at least conceptually) separated, mobile devices provide a platform allowing the creation of a wide range of standalone musical instruments. The combination of different types of sensors, a microphone, a speaker and a powerful small computer in a battery powered single entity makes this possible. Moreover, the fact that this kind of device is mass produced guaranties a certain level of robustness which is not always the case of DIY (Do It Yourself) interfaces and instruments.

Carried by a huge market, mobile devices evolve very fast, and today's smartphones and tablets are quite different from the ones available a couple of years ago [7]. While only simple instruments and basic interactions could be implemented then, today's devices give access to a huge amount of computational power, low-latency touch screens allowing on the order of ten simultaneous touches, and good quality ADC/DACs.

However, mobile devices were never designed to be used as musical instruments and they are often missing crucial elements to be as expressive and versatile as professional musical instruments [8]. For instance, the lack of force sensitivity on touch screens narrows the range of possible interactions, and makes performance with specific classes of instruments such as percussion and plucked string instruments less intuitive. For example, striking force must be substituted by some other dimension such as the y coordinate in a strike area.

Park et al. addressed this issue a few years ago and proposed a solution using the built-in accelerometer of the device and a foam padding [9]. While this solution is very self-contained as it uses only the built-in sensors of the device, it presents several limitations diminishing the range of practical applications (e.g., no multi-touch support, sensitivity to table/support vibrations, no automatic re-calibration, limited sampling rate, etc.).

Some of the most recent generations of devices such as the *iPhone 6* provide basic multi-touch force detection on the screen ("3D Touch"). [1] This feature has already been exploited by some companies such as ROLI with its *Noise* app [2] to create expressive musical instruments. Unfortunately, this technology is not yet available on larger screen devices (tablets, etc.) that provide a better interface to control certain type of instruments such as percussion [10]. Instead, tablet manufacturers currently favor the use of force sensitive pencils [3] [4] that provide a simpler solution to this problem. Also, the "3D Touch" technology of the *iPhone 6* has some limitations. While it can provide very accurate data in the case of a continuous touch event ("after-touch"), its useability for deriving the *velocity* of a strike gesture on the screen is very limited. This makes it practically unusable to control percussion or plucked string instruments where the attack is very sharp.

In this paper, we introduce NUANCE: a device adding high quality multi-touch low-latency force detection to the *iPad* touch screen, fast and accurate enough to be suitable for deriving striking velocity. NUANCE is based on four force sensitive sensors placed on each corner at the back of the device. It communicates with the *iPad* using its *audio jack* connector through a purely analog system streaming the sensor data as an audio signal. This ensures a fast data rate (up to the audio bandwidth of the *iPad*, nominally 20 kHz) as well as a high sample resolution (bit depth).

After describing the hardware implementation of NUANCE, we demonstrate how it can be used to design musical instruments. We then provide a series of examples, evaluate its performance and discuss future applications and improvements.

[1] http://www.apple.com/iphone-6s/3d-touch/ All the URLs presented in this paper were checked on 2016-07-11.
[2] https://www.roli.com/products/noise.
[3] http://www.apple.com/apple-pencil/.
[4] https://www.microsoft.com/surface/en-us/accessories/pen.

2. HARDWARE

The case of NUANCE is made out of wood (plywood and birch) and black laser cut acrylic (see figure 1). The current version was designed for the *iPad Air 2* [5] but it is also compatible with the *9.7 in iPad Pro*. [6]

An FSR (Force Sensitive Resistors) [7] is placed under each corner of the *iPad* (see Figure 2). The FSRs are covered with a thin (1/4 in) piece of foam whose rigidity was chosen to offer a good compromise between responsiveness and damping [10]. The foam is used to cushion the strikes of the performer and also to give some slack to the *iPad* during continuous push gestures (after-touch).

The signals from the different FSRs is sent to the *iPad* using amplitude modulation (AM). Each force signal controls the gain of its own analog oscillator. The oscillators are very simple based on a *555 timer* (see Figure 3). This kind of circuit doesn't generate a pure sine wave but it is straightforward to efficiently isolate each carrier wave during the demodulation process described in §3 (reducing the dynamic range of the signal is not a problem since it is pretty large, thanks to the audio ADC).

The frequency of each oscillator is different and is controlled by $R1$ and $C1$. Frequencies (2, 6, 10 and 14 kHz) are spread across the bandwidth of the line input of the *iPad* (assuming that the sampling rate of the target app is at least 44.1 kHz). Since we're not carrying an audio signal, and since the sharpest attack we could achieve by tapping the screen was longer than 10ms (corresponding to a bandwidth less than 100Hz), we don't have to worry about sidebands. Moreover, the demodulation technique used on the *iPad* (see §3) significantly reduces the risk of sidebands contaminating neighboring signals.

The FSRs were calibrated to output their maximum value when a weight of approximately 400 grams is applied on the touch screen. This was set empirically to provide the best feeling to our taste but this value (that should remain reasonable to not damage the screen) can be easily adjusted by a small potentiometer mounted on the circuit board. Since the *iPad* itself applies some weight to the FSRs, the minimum value of the range is constantly adjusted on the software side (see §3).

The output of the different oscillators is mixed and sent to the *iPad* using the line input pin of the *audio jack* input.

The system is powered with an external 5 V power supply and connects to the *iPad* using a 3.5 mm (1/8 inch) headset (four-pole) audio jack. The output signal is routed to a 1/4 inch stereo audio jack mounted on the side of NUANCE.

The circuits were chosen to be as simple as possible to reduce the cost of NUANCE to less than $30. Variations of the sine oscillators (stability of the frequency, purity of the sine wave, etc.) are easily compensated on the software side.

[5] `http://www.apple.com/ipad-air-2/`.

[6] `http://www.apple.com/ipad-pro/`.

[7] The FSRs used for the device are *Interlink Electronics FSR 400*: `http://www.interlinkelectronics.com/FSR400.php`.

3. SOFTWARE

The amount of force applied to each FSR of NUANCE is carried by different sine waves to the *iPad* using its single audio analog input. Four band-pass filters isolate the signal of each sine wave (see Figure 4). Their bandwidth is big enough to accommodate the variations of the frequencies of the simple sine oscillators described in §2. The amplitude of the output signal of each filter is extracted and corresponds to the force measured by each FSR of NUANCE.

The DSP (Digital Signal Processing) portion of the different apps compatible with NUANCE that are presented in §4 is implemented using the FAUST [8] programming language [11]. The audio process is wrapped in a `C++` library using MOBILEFAUST [12] making it accessible to the higher level `Objective-C` layers of the app.

The force information extraction system described in the previous paragraph is implemented as a single FAUST function that is executed in the same audio callback function as the synthesizer that it is controlling. In other words, the force data from the FSRs are acquired and are controlling the synthesizer at the audio rate.

Touch events (including the (x, y) position on the screen) retrieved in the `Objective-C` layer are sent to the FAUST DSP object using the API provided by MOBILEFAUST. These data are compared with the data provided by the FSRs to associate a force signal to a specific touch event on the screen (see Figure 4). If the touch event was just initiated, the force is converted into a velocity proportional to the instantaneous force at the beginning of the touch. If the touch event persists, then the force is converted into a series of after-touch events.

As mentioned in §2, the *iPad* itself applies a certain amount of weight to the FSRs. Depending on its position in the case or the level of inclination of the table where NUANCE is installed, this value can vary a little bit. To make sure that the range of the FSRs is always accurate, it is readjusted when there are no fingers touching the screen. If there are several simultaneous active touches (multi-touch) on the screen, a simple triangulation algorithm compares the force level at each FSR with the X/Y position of the touch on the screen to associate a velocity or an after-touch event to it. Obviously, the more simultaneous touches on the screen, the harder it becomes for the system to differentiate independent forces. We find that the system is very accurate in the case of two simultaneous touches, but the force distribution tends to become more uniform if more touches are engaged. However, we find that this is not an issue for many types of percussion and plucked-string instrument control.

4. EXAMPLES

While it would be quite easy to write an app to use NUANCE as a MIDI controller, we like the idea of creating standalone musical instruments taking full advantage of the possibilities offered by this system. For this reason, the different apps that we created and that are compatible with NUANCE target specific instruments. However, we hope to

[8] `http://faust.grame.fr`.

Figure 1. Global view of NUANCE.

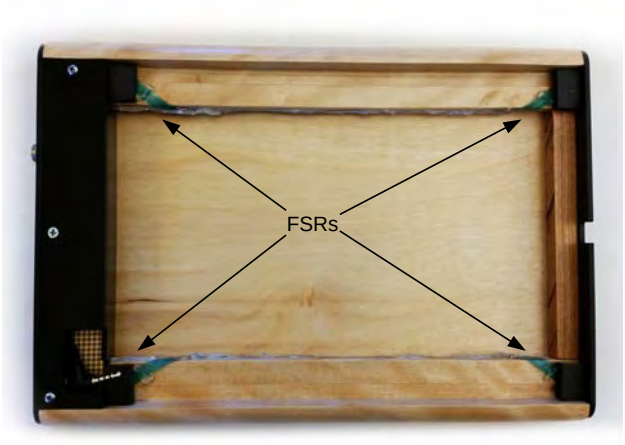


Figure 2. Top view of NUANCE without the *iPad*.

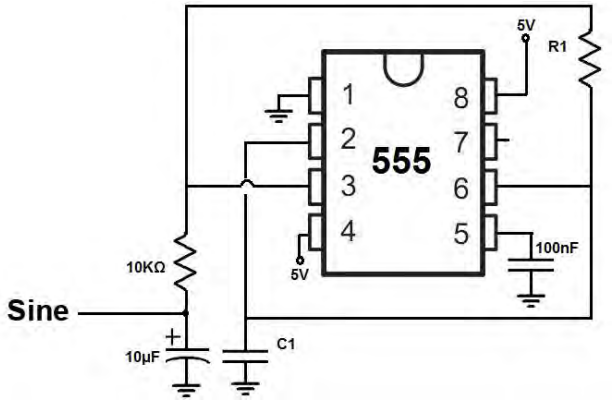


Figure 3. Circuit diagram of one of the simple sine oscillators used in NUANCE.

create a platform (see §6) to facilitate the design of multiple kinds of instruments, based on a mobile device and some hardware augmentation where several instruments would be accessible through a single app.

While NUANCE would work well with a wide range of screen interfaces (piano keyboards, isomorphic keyboards, etc.), we mostly focused percussion instruments so far. The different apps that we created implement one or several drums represented by rectangular regions on the touch screen. The app presented in Figure 5 has three different zones, each controlling a different drum. Each offers a physical representation of the virtual instrument (striking on the edges sounds different than striking at the middle, etc.). The drum synthesizers are all based on modal physical models [13] which strengthens the link between the physical and virtual parts of the instrument, increasing its overall "physical coherence". The physical models were implemented in FAUST using the FAUSTSTK [14].

The multi-touch capabilities of NUANCE allow to simultaneously strike two drums with different velocities. The after-touch information can also be used to interact with the resonance time (T60) of the virtual drums. This results in a highly expressive instrument [9].

5. EVALUATION/DISCUSSION

The "most standard" way to connect an external music controller to the *iPad* is by using MIDI through the lightening connector. While this solution works well for most basic applications (e.g., a MIDI keyboard triggering events in a synthesizer), the limited bandwidth and bit depth, as well as the jitter in the latency of MIDI, can be problematic for applications requiring a high rate of data and precise synchronization [15].

The idea of using the line-input of the *audio jack* plug of mobile devices to send data to it has already been exploited a lot. Various commercial products such as credit-

[9] `https://ccrma.stanford.edu/~rmichon/nuance/` presents a series of demo videos of NUANCE.

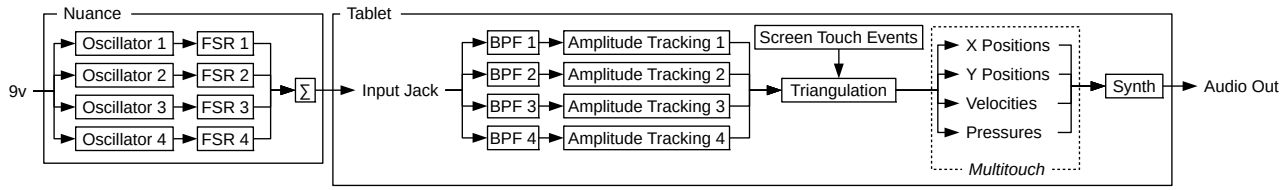


Figure 4. Overview of NUANCE.

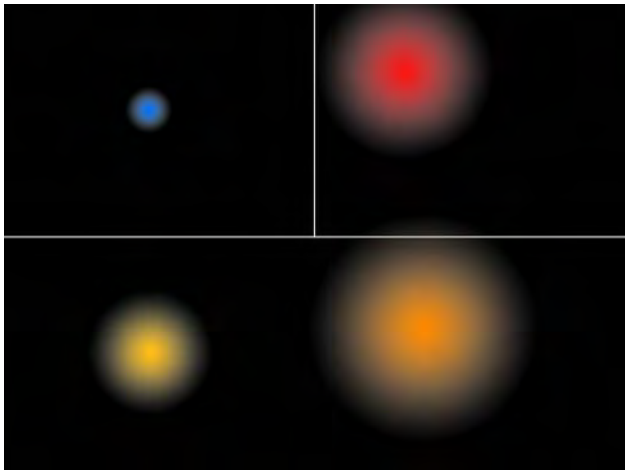


Figure 5. Screenshot of one of the percussion apps compatible with NUANCE. The blurry circles represent the strikes and their velocities (diameter).

card readers, like *Square*, use this technique, but this idea has also been used by the DIY community to send sensor data. For example, [16] is a simple modem that uses the *audio jack* connector of *Android* and *iOS* devices to transmit digital data. Its main limitation is its very small bandwidth (30 bytes/sec). [17] uses a different paradigm where the analog signals are multiplexed and sent one after the other.

Our approach described in the three previous sections is less versatile and more "low-tech" but it allows to stream the signals from four sensors to the *iPad* using the audio bandwidth. The consistency and rate of the information remain constant, greatly simplifying its synchronization with the sound synthesizer. It is also much cheaper and easier to build from scratch.

Its main disadvantage is that the demodulation technique (see §3) that it uses on the mobile device to retrieve the sensor data is rather computationally expensive. However the huge power of modern mobile devices compensates for this, and running the various band-pass filters, amplitude trackers, and the triangulation function presented in Figure 4 only takes 4% of the resources of the CPU on an *iPad Air 2*.

A more general limitation of our system is the fact that force can't be accurately measured on more than two independent simultaneous touches on the screen. We have not found this to be an issue in musical-performance applications to date. Perhaps only a built-in technology, such as 3D Touch, can efficiently resolve this issue. The fact that the most recent versions of *iPhones* address this problem leads us to think that larger devices such as the *iPad* will follow this trend too in future models. As mentioned in the introduction, it seems that tablet manufacturers prefer to settle for a force-sensitive pencil for now, which does not provide multi-touch force. Also, while the technology used by *Apple iPhone 6s* is fully multi-touch, it is not as fast as the method presented in this paper, and it can't be used for example to accurately detect the velocity of striking gestures.

6. FUTURE WORK

While the current version of NUANCE is quite stable and works well in its current state, the ability to power it through the *audio jack* using the technique described in [18], and used in [17], would be a good way to improve it.

Even though the technology used by NUANCE to communicate with the *iPad* is compatible with any other kind of mobile device (both *iOS* and *Android*), it is hard to find a design that adapts well to the size of any device in the same category. We would like to find a solution to this problem so that NUANCE can work with any types of *iPads* and *Android* tablet.

More generally, NUANCE was made in the frame of a larger scale project investigating the idea of "mobile device augmentation" and "hybrid lutherie" where our goal is to create a toolbox/platform to help design this kind of musical instrument.

7. CONCLUSIONS

NUANCE is a new device adding force-touch and velocity detection to the *iPad* touch screen. It communicates with the *iPad* using the *audio jack* input. Force signals are sent at an audio rate using analog amplitude modulation. NUANCE provides a high level of sensitivity and responsiveness by only using analog components. It costs less than $30 to make.

While NUANCE associates a different force value to simultaneous touch events happening on the screen, its ability to precisely distinguish the amount of force applied by different fingers decreases as the number of touches increases. This is probably its main limitation.

We think that adding this extra force dimension to the touch screen of the *iPad* significantly increases its expressive potential for controlling any musical instrument.

Acknowledgments

I would like thank my colleagues at CCRMA (especially

Fernando Lopez-Lezcano) for supporting the large amount of noise pollution that I generate.

8. REFERENCES

[1] A. Tanaka, "Mobile music making," in *Proceedings of the 2004 conference on New interfaces for musical expression (NIME04)*, National University of Singapore, 2004.

[2] G. Schiemer and M. Havryliv, "Pocket gamelan: tuneable trajectories for flying sources in mandala 3 and mandala 4," in *Proceedings of the 6th International Conference on New Interfaces for Musical Expression (NIME06)*, Paris, France, 2006.

[3] G. Essl, G. Wang, and M. Rohs, "Developments and challenges turning mobile phones into generic music performance platforms," in *Proceedings of the Mobile Music Workshop*, Vienna, Austria, May 2008.

[4] G. Essl, M. Rohs, and S. Kratz, "Use the force (or something) - pressure and pressure-like input for mobile music performance," in *Proceedings of the New Interfaces for Musical Expression Conference (NIME-10)*, Sydney, Australia, June 2010.

[5] G. Wang, J. Oh, and T. Lieber, "Designing for the ipad: Magic fiddle," in *Proceedings of the International Conference on New Interfaces for Musical Expression*, Oslo, Norway, May 2011.

[6] G. Wang, "Ocarina: Designing the iphones Magic Flute," *Computer Music Journal*, vol. 38, no. 2, pp. 8–21, Summer 2014.

[7] T. Martin, "The evolution of the smartphone," July 2014, http://pocketnow.com/2014/07/28/the-evolution-of-the-smartphone. Accessed: 2016-04-09.

[8] R. Michon, J. O. Smith, M. Wright, and C. Chafe, "Augmenting the ipad: the bladeaxe," in *Proceedings of the International Conference on New Interfaces for Musical Expression*, Brisbane, Australia, July 2016.

[9] T. H. Park and O. Nieto, "Fortissimo: Force-feedback for mobile devices," in *Proceedings of the International Conference on New Interfaces for Musical Expression*, KAIST, Daejon, Korea, May 2013.

[10] Z. Ren, R. Mehra, J. Coposky, and M. C. Lin, "Tabletop ensemble: Touch-enabled virtual percussion instruments," in *Proceedings of the ACM SIGGRAPH Symposium on Interactive 3D Graphics and Games (I3D 12)*, Costa Mesa, California, March 2012.

[11] Y. Orlarey, D. Fober, and S. Letz, "An algebra for block diagram languages," in *Proceedings of the International Computer Music Conference (ICMC-02)*, Gothenburg, Sweden, 2002.

[12] R. Michon, J. O. Smith, and Y. Orlarey, "Mobilefaust: a set of tools to make musical mobile applications with the faust programming language," in *Proceedings of the Linux Audio Conference (LAC-15)*, Mainz, Germany, April 2015.

[13] J.-M. Adrien, "The missing link: Modal synthesis," in *Representations of Musical Signals*. Cambridge, USA: MIT Press, 1991, ch. The Missing Link: Modal Synthesis, pp. 269–298.

[14] R. Michon and J. O. Smith, "Faust-stk: a set of linear and nonlinear physical models for the faust programming language," in *Proceedings of the 14th International Conference on Digital Audio Effects (DAFx-11)*, Paris, France, September 2011.

[15] G. Loy, "Musicians make a standard: The midi phenomenon," *Computer Music Journal*, vol. 9, no. 4, pp. 8–26, Winter 1985.

[16] SwitchScience, "Audio jack modem for iphone and android," 2016, https://www.switch-science.com/catalog/364/. Accessed: 2016-04-09.

[17] S. Verma, A. Robinson, and P. Dutta, "Audiodaq: Turning the mobile phones ubiquitous headset port into a universal data acquisition interface," in *Proceedings of the Conference on Embedded Networked Sensor Systems (SenSys)*, Toronto, Ontario, November 2012.

[18] Y.-S. Kuo, T. Schmid, and P. Dutta, "Hijacking power and bandwidth from the mobile phones audio interface," in *Proceedings of the International Symposium on Low Power Electronics and Design (ISLPED)*, Austin, Texas, 2010.

FAUCK!! HYBRIDIZING THE FAUST AND CHUCK AUDIO PROGRAMMING LANGUAGES

Ge Wang
CCRMA, Stanford University
ge@ccrma.stanford.edu

Romain Michon
CCRMA, Stanford University
rmichon@ccrma.stanford.edu

ABSTRACT

This paper presents a hybrid audio programming environment, called FAUCK, which combines the powerful, succinct Functional AUdio STream (FAUST) language with the strongly-timed CHUCK audio programming language. FAUCK allows programmers to on-the-fly evaluate FAUST code directly from CHUCK code and control FAUST signal processors using CHUCK's sample-precise timing and concurrency mechanisms. The goal is to create an amalgam that plays to the strengths of each language, giving rise to new possibilities for rapid prototyping, interaction design and controller mapping, pedagogy, and new ways of working with both FAUST and CHUCK. We present our motivations, approach, implementation, and preliminary evaluation. FAUCK is open-source and freely available.

1. INTRODUCTION

A variety of computer music programming languages exist for the same reason there are many different types of tools: each is well-suited to different types of tasks, and speaks to different aesthetic and pragmatic preferences of the programmer. FAUST and CHUCK are two audio programming languages that effectively illustrate this point. FAUST (Functional AUdio STream) [1–4] embraces a declarative and functional paradigm, is succinct, tailored to expressively describe low-level digital signal processing (DSP) algorithm, and generates optimized, efficient synthesis modules. CHUCK, [5, 6], on the other hand, is imperative, designed around a notion of temporal determinism that includes sample-synchronous timing and concurrency (called *strongly-timed*), tailored for precise control, readability, and an on-the-fly rapid-prototyping mentality [7]. Yet, they share the general goal of sound synthesis for musical applications and both are text-based languages (e.g., not graphical patching).

What happens when one combines these two languages? More to the point, can these languages be combined in such a way to take advantage of the respective strengths of both? Furthermore, that the two languages seem vastly different in syntax, semantics, and personality is all the more reason to explore their intersections (i.e., why try to combine two things are already similar?). Might their profound differences give rise to something different from either alone? Such curiosities provide the primary motivation for our exploration in hybridizing FAUST and CHUCK.

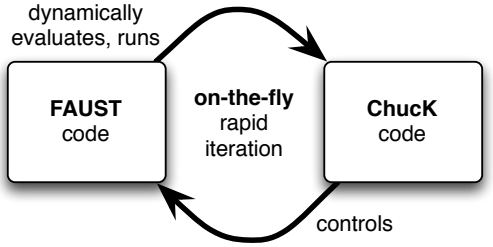


Figure 1. New FAUCK model.

FAUCK is a tight integration between FAUST and CHUCK that attempts to infuse their respective characteristics, reinforcing and even augmenting each language with aspects of the other (Figure 1). It is designed such that programmers can embed and evaluate FAUST code directly from within a CHUCK program, take full advantage of CHUCK's sample-synchronous time-based control and concurrency, and do all of this on-the-fly. At the same time, this makes available to CHUCK the entire existing body of FAUST programs, ready to be used for synthesis and interaction design. In combination, FAUCK provides a different hybridized way to rapidly prototype sample-precise audio synthesis code in FAUST and control it precisely using CHUCK.

2. RELATED WORK

2.1 Existing Paradigm

Thanks to its architecture system [8], FAUST can be used to easily build custom DSP modules that generate or process samples as part of a larger host. The traditional paradigm of incorporating FAUST into computer music systems involves a pipeline that 1) generates C++ code from FAUST code, 2) compiles into a plug-in, and 3) runs as part of a host software system [1] (e.g., *PureData* [2], *Max/MSP* [3], *SuperCollider* [4], etc.). In this regime, there is a new plug-in created from any given FAUST program (Figure 2). Such a system currently exists for compiling a given FAUST program into a CHUCK *chug-in* [9].

[1] http://faust.grame.fr/Documentation/ contains an exhaustive list of the FAUST architectures.
[2] https://puredata.info/.
[3] https://cycling74.com/products/max/.
[4] http://supercollider.github.io/.

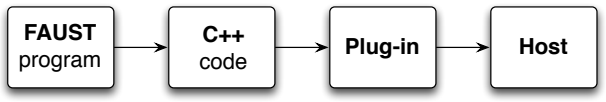


Figure 2. Traditional FAUST code to plug-in model.

2.2 Dynamic FAUST Extensions

Modules created using the technique described in §2.1 are static and cannot be modified once they are compiled. `libfaust` [10] which is part of the FAUST2 distribution [5] allows to embed the FAUST compiler in any program written in C++ to dynamically generate DSP modules on the fly (see §4).

Since its introduction in 2012 `libfaust` has been used in various projects to integrate the FAUST compiler to existing platforms or to create new one. The FAUST backend [11] of the *Pure* programming language [12] that predates the creation of `libfaust` was used as the basis for the design of this tool.

`libfaust` was used for the first time in *PureData* within a dynamic *external* that can be turned into any DSP object by modifying the FAUST code linked to it [13]. Similarly `faustgen` [14] is a dynamic *MaxMSP* external borrowing the paradigm used in `gen` [15] to create custom unit generators. *CSound* [6] now hosts a dynamic *opcode* functioning the same way than the two previous examples [16]. `libfaust` has also been used under different forms within the *Web Audio API* [17] to create custom dynamic *nodes* [18, 19]. More recently, FAUST has been integrated to the *JUCE* [7] platform [20] and the *Processing* [8] programming language.

Finally, `libfaust` is at the core of the FAUSTLIVE "just in time" FAUST compiler [21] allowing to write FAUST code in a text file and generating the corresponding standalone audio application almost instantly.

3. THE FAUCK APPROACH

3.1 Using FAUCK

FAUST objects can be used easily in any CHUCK code through a *chugin* called `Faust`. For example, a new FAUST unit generator (e.g., an audio DSP effect that takes an input from CHUCK) can be declared as follow:

```
adc => Faust foo => dac;
```

In the case where `foo` would be a synthesizer, the `adc` would be ignored and we could simply write:

```
Faust foo => dac;
```

Any FAUST program can be associated with `foo` and dynamically evaluated by calling the `eval` method.

```
foo.eval(`process=osc(440);`);
```

For brevity and convenience, several common libraries (`music.lib`, `filter.lib`, `oscillator.lib`, `effect.lib`, `math.lib`) are, by default, automatically imported by FAUCK. Furthermore, note the use of the backtick (`` ` ``) to delineate the inline FAUST code – this removes the need to manually escape single and double quotation marks used in the FAUST code.

Alternately, the same object can load a FAUST program from the file system by invoking `compile` and providing a path to a FAUST `.dsp` file:

```
foo.compile("osc.dsp");
```

Next, the `v` method can be called at anytime to change the value of a specific parameter defined on the FAUST object that is specified by its path (`v` stands for "value"; we chose this abbreviation in anticipation that most program will invoke this method often). For example, here we create a sine wave oscillator whose only parameter is its frequency (`freq`) and we set it to 440Hz:

```
foo.eval(`
    frequency = nentry("freq",
        200,50,1000,0.01);
    process = osc(frequency);
`);
foo.v("freq",440);
```

Finally, the `dump` method can be called at any time to print a list of the parameters of the FAUST object as well as their current value. This is useful to observe large FAUST programs that have a large number of parameters in complex grouping paths. Programmers can also directly copy the path of any parameter to control for use with the `v` method.

3.2 Examples

3.2.1 A Simple Example

The following example puts together the different elements given in §3.1 by implementing a simple sine wave oscillator (specified in FAUST) whose frequency and gain are randomly changed every 100ms (controlled in CHUCK).

```
// connect a Faust object to ChucK dac
Faust foo => dac;
// evaluate
foo.eval(`
    frequency = nentry("freq",
        200,50,1000,0.01) : smooth(0.999);
    gain = nentry("gain",
        1,0,1,0.01) : smooth(0.999);
    process = osc(frequency)*gain;
`);
// ChucK time loop
while( true ){
    // control frequency
    foo.v("frequency", Math.random2f(50,800));
    // control gain
    foo.v("gain", Math.random2f(0,1));
    // advance time
    100::ms => now;
}
```

3.2.2 An Advanced Example

Making use of CHUCK's sample-precise timing and concurrency mechanisms, it is straightforward to mix CHUCK

[5] `https://sourceforge.net/p/faudiostream/code/ci/faust2/tree/`.
[6] `http://www.csounds.com/`.
[7] `https://www.juce.com/`.
[8] `https://processing.org/`.

unit generators with FAUST objects to create hybrid elements. In the following example, a string physical model implemented in an external FAUST file is filtered by a *crybaby* effect evaluated in the CHUCK file and declared in `effect.lib` (FAUST library). The `wah` parameter of the *crybaby* effect is modulated by an *LFO*[9] declared as a CHUCK object. The string physical model is controlled in concurrent CHUCK shreds, spawned through the `spork` operator.

```
// instantiate and connect 2 Faust modules
Faust string => Faust cryBaby => dac;
// LFO using ChucK UGen
SinOsc LFO => blackhole; 6 => LFO.freq;

// load FAUST program; map to Faust object
string.compile("string.dsp");
// evaluate code; crybaby from effect.lib
cryBaby.eval(`process = crybaby_demo;`);

// generates random notes
fun void notes(){
    while( true ){
        // new note
        string.v("gate",0);
        10::ms => now;
        string.v("gate",1);
        // with random frequency
        string.v("freq",
        Math.random2f(80,800) );
        100::ms => now;
    }
}
// modulates the cry baby with the LFO
fun void lfoWah(){
    while( true ){
        cryBaby.v(
            "/CRYBABY/Wah_parameter",
            (LFO.last()*0.5+0.5) );
        1::samp => now; // every sample!
    }
}

spork ~ notes();
spork ~ lfoWah();
while( true ){
    10::ms => now;
}
```

4. IMPLEMENTATION

FAUCK is implemented as a *chugin* (CHUCK plugin), simply named `Faust`. The chugin, when installed, shows up as the FAUST unit generator in CHUCK, and can be used in any number or configuration from CHUCK (as shown in the code example above). FAUCK internally manages the interface between CHUCK and the just-in-time FAUST compiler. Each instance of the `Faust` unit generator maintains a map of parameters indexed on the full parameter path, which enables real-time look-up and direct manipulation of the named parameters.

FAUCK is written in C++ and is made possible by `libfaust`, an embedded version of the FAUST compiler, capable of generating *LLVM* bitcode (*LLVM IR*) instead of C++. `libfaust` invokes the *LLVM* compiler with the bitcode to emit into an efficient binary format that can be run dynamically [10].

[9] *Low Frequency Oscillator*.

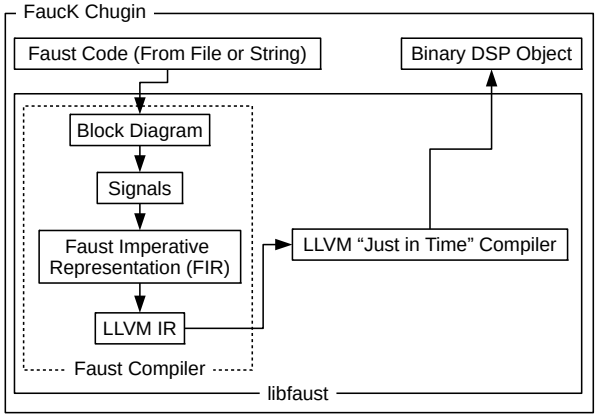


Figure 3. Overview of the FAUCK *chugin*.

Real-time performances seem promising. The advanced example in the previous section runs below 7% CPU utilization on a MacBook Pro from 2012 (this includes the baseline overhead of the CHUCK virtual machine). For comparison, a similar example, missing the `notes()` function, fully implemented in FAUST, and compiled as a standalone *CoreAudio* application, runs at approximately 5% on the same computer.

5. CONCLUSIONS AND FUTURE WORK

As it turns out (and as we had hoped), the pronounced differences between FAUST and CHUCK actually make it easier to articulate a useful intersection between the two languages. FAUST code tends to operate at the within-unit generator DSP level, whereas CHUCK's unique strength lies in the strongly-timed, concurrent control of unit generators. This makes for easy division of labor in FAUCK. Furthermore, it is straightforward in this model to mix FAUST modules with existing CHUCK unit generators.

More generally, FAUCK provides a few unique benefits:

- FAUCK combines the control capabilities of CHUCK to the efficiency and concise, expressive DSP programming of FAUST.
- FAUST has no scheduler/control system — something CHUCK was specifically designed for (making for example, polyphonic FAUST objects is now easy with FAUCK).
- The `Faust` modules provides seamless integration with CHUCK unit generators, enabling a new type of rapid prototyping and experimentation in FAUST, CHUCK, or in tandem.
- Overall, FAUCK provides a new way to work with FAUST, while expanding CHUCK's synthesis capabilities to include the large and growing body of FAUST code. FAUCK presents a clear deterministic all-in-one place delineation of both FAUST and CHUCK code, which can potential benefit both research and classroom settings.

For future work, we'd like to continue experimenting with features in FAUCK to further facilitate this hybridization. For example, while it's possible for CHUCK code to control all parameters in a FAUST program, regardless of the type of UI defined for the parameter, it would be convenient if FAUCK can provide functionality to auto-generate miniAudicle user interfaces (MAUI) [22] from any FAUST code. Also, since both FAUST and CHUCK are text-based, it would be intriguing to further deepen the intersection with dynamically- or self-generating FAUST code from within CHUCK. Also, we are beginning to apply FAUCK in computer music pedagogical settings, as well as towards DSP-based physical modeling and computer-mediated instruments design for laptop orchestras.

FAUCK is open-source and is part of the `ChuGin` repository: `https://github.com/ccrma/chugins`.

Acknowledgments

We thank our colleagues at CCRMA and GRAME for their suggestions and support.

6. REFERENCES

[1] Y. Orlarey, S. Letz, and D. Fober, *New Computational Paradigms for Computer Music*, Paris, France, 2009, ch. FAUST : an Efficient Functional Approach to DSP Programming.

[2] Y. Orlarey, D. Fober, and S. Letz, "An algebra for block diagram languages," in *Proceedings of the International Computer Music Conference*, 2002.

[3] R. Michon and J. O. Smith, "Faust-stk: A set of linear and nonlinear physical models for the faust programming language," in *Proceedings of the 14th International Conference on Digital Audio Effects*, 2011.

[4] R. Michon, J. O. Smith, and Y. Orlarey, "Mobilefaust: a set of tools to make musical mobile applications with the faust programming language," in *Proceedings of the Linux Audio Conference*, 2015.

[5] G. Wang, "The chuck audio programming language," Ph.D. dissertation, Princeton University, 2008.

[6] G. Wang, P. R. Cook, and S. Salazar, "Chuck: A strongly timed computer music language," *Computer Music Journal*, vol. 39, no. 4, pp. 10–29, 2015.

[7] G. Wang and P. R. Cook, "On-the-fly programming: Using code as an expressive musical instrument," in *New Interfaces for Musical Expression*, 2004.

[8] D. Fober, Y. Orlarey, and S. Letz, "Faust architectures design and osc support," in *Proceedings of the 14th International Conference on Digital Audio Effects (DAFx-11)*, Paris, France, September 2011.

[9] S. Salazar and G. Wang, "Chugens, chubgraphs, and chugins: 3 tiers for extending chuck," in *Proceedings of the International Computer Music Conference*, 2012.

[10] S. Letz, D. Fober, and Y. Orlarey, "Comment embarquer le compilateur faust dans vos applications?" in *Proceedings of the Journées de l'Informatique Musicale*, Paris, France, May 2013.

[11] A. Gräf, "An llvm bitcode interface between pure and faust," in *Proceedings of the Linux Audio Conference (LAC-11)*, Maynooth, Ireland, May 2011.

[12] A. Gräf, "Signal processing in the pure programming language," in *Proceedings of the Linux Audio Conference (LAC-09)*, Parma, Italy, April 2009.

[13] A. Gräf, "pd-faust: An integrated environment for running faust objects in pd," in *Proceedings of the Linux Audio Conference (LAC-12)*, Stanford, California, April 2012.

[14] "Faustgen," 2012. [Online]. Available: http://faust.grame.fr/news/2012/12/11/faustgen.html

[15] "Gen documentation," 2016. [Online]. Available: https://cycling74.com/wiki/index.php?title=gen~_For_Beginners

[16] V. Lazzarini, "Faust programs in csound," *Revue Francophone d'Informatique Musicale*, no. 4, Fall 2014.

[17] "The web audio api," 2015. [Online]. Available: https://www.w3.org/TR/webaudio/

[18] S. Denoux, Y. Orlarey, S. Letz, and D. Fober, "Compose with faust in the web," in *Proceedings of the Web Audio Conference*, Paris, France, January 2015.

[19] S. Letz, S. Denoux, Y. Orlarey, and D. Fober, "Faust audio dsp language in the web," in *Proceedings of the Linux Audio Conference (LAC-15)*, Mainz, Germany, April 2015.

[20] O. Larkin, "Using the faust dsp language and the libfaust jit compiler with juce," in *Proceedings of the JUCE Summit*, London, UK, November 2015.

[21] S. Denoux, S. Letz, Y. Orlarey, and D. Fober, "Faustlive: Just-in-time faust compiler... and much more," in *Proceedings of the Linux Audio Conference (LAC-12)*, Karlsruhe, Germany, April 2014.

[22] S. Salazar, G. Wang, and P. R. Cook, "miniaudicle and chuck shell: New interfaces for chuck development and performance." in *Proceedings of the International Computer Music Conference*, 2006.

Teaching Audio Programming using the Neonlicht Engine

Jan-Torsten Milde
Fulda University of Applied Sciences, Digital Media Working Group, CS Department
milde@hs-fulda.de

ABSTRACT

In the following text, we describe the ongoing development of an efficient, easy to use, scalable synthesizer engine: Neonlicht. Neonlicht has been developed with the objective to be used as a teaching tool as part of a study program in digital media. With the system we like to improve the quality of teaching in the subject area of audio programming.

1. INTRODUCTION

In the following text, we describe the ongoing development of an efficient, easy to use, scalable synthesizer engine: Neonlicht. Neonlicht has been developed with the objective to be used as a teaching tool as part of a study program in digital media. With the system we like to improve the quality of teaching in the subject area of audio programming.

A rather large didactic problem is found, when looking at the current situation of many students of digital media. On the one hand students tend to have a high affinity towards using the the computer as a media system. Indeed the computer is used to consume the entire spectrum of digital media. At the same time though the access to the computer as a system for software *development* becomes increasingly more difficult. Similar perceptions relating to work and study behaviour are now observable in many study programs, which is at least the author's estimation after a number of discussions with teaching colleagues in a number of countries.

2. SCHEDULE OF A CURRENT AUDIO PROGRAMMING COURSE

Our starting point for the instructional design of the current course was the insight that students have to be carefully guided in order to get access to the complex topics of audio programming. A larger fraction of them displays deficits in both programming skills, as well as in mathematical foundations. Furthermore using mathematics is often referred to as being "uncreative" and leads to immediate "learning rigor mortis".

As a basic precondition our course had to be constructed in a *practice oriented* way: the acquisition of knowledge through practical application often leads to better results in understanding the instructional material.

The introductory phase of the course was started by building (or to be more accurate: by assembling) physical synthesizers: for this two Moog [1] systems were purchased. This entry phase was very successful. Students loved experimenting with the synthesizers. For the subsequent analysis of the Moog's functionality and its simulation on the computer Pure Data was used (see [1]). Again, the entry threshold to this technology is relatively low, since software development is supported by a fairly intuitive graphic user interface.

Figure 1. Neonlicht runs on a Raspberry PI 3 (the image displays it's predecessor, the Raspberry 2).

Moreover, the author bought two CCRMA Satellite Systems [2] at an ICMC/SMC 2014 workshop in Athens. In these systems, Pure Data runs on a Raspberry 1. Via an additional Arduino it becomes possible to connect external hardware components to the system and then use them as hardware controls for Pure Data. The students were therefore able to run larger parts of their Pure Data patches on this hardware setup.

The course then continued with a step by step introduction to the most common synthesis methods. The theory parts were supported by corresponding practical exercises.

In the final phase of the course each student had to define, plan and implement a freely chosen individual project. The outcomes were very pleasing: amongst other things the students developed a guitar effect processor with PureData running on a Raspberry Pi 2, a gesture-based musical instrument utilizing WII controllers and an 8bit sound machine.

[1] http://www.moogmusic.com/products/Werkstatt
[2] https://ccrma.stanford.edu/ eberdahl/Satellite/

Overall, the results were positive, but at the same time demonstrated the problems very clearly. The students were very disappointed of the lack in practicality of the systems used. They felt that the development work was in many ways a to high challenge, as the system did not support error tracking and debugging well, modularization is not well supported when developing larger systems, the performance of the systems is very limited and only a rather poor system integration could be achieved.

In total, they had the impression that you can only build simple test and experimental systems, but when it comes to building "real", larger systems, these tools are not suitable.

3. NEONLICHT: REQUIREMENTS

The starting point for the development of Neonlicht, an engine for the efficient programming of synthesizers and audio processing components, thus was the insight, that although there is a variety of systems for audio programming, that none of the proposed approaches met the educational requirements of the current course. In sum, with Neonlicht we therefore try to implement the following requirements:

- simplicity of the underlying program structure
- comparability with existing standard audio applications (Pure Data, STK ([2]), ChucK ([3]), SuperCollider ([4]) etc.), allowing for an easy transition
- a reduction (!) in the number of audio functions on the didactically relevant
- flexibility and easy expandability of the basic functions
- ease of installation of the system
- avoiding if possible all programming constructs that go beyond basic knowledge of object oriented programming (in Java) (especially use of pointers and memory management)
- low resource consumption, enabling the system to run on simple and cheap hardware
- high (professional) quality of synthesis and audio output
- a clear logical and physical separation of engine and control
- maximum portability, at least for Linux/Unix based systems

3.1 Using the Raspberry Pi 3

It is a key technical requirement in the development of the Neonlicht engine to make it run on a current Raspberry Pi 3 (see [5]). In its third generation this micro-computer is so powerful, that the students will be able to programm and experiment directly on the machine. In combination with a low-cost USB audio card and a standard MIDI controller musical instruments can be developed that can even be used in live situations.

Despite of the relatively high performance of the Raspberry 3, it is still necessary to implement Neonlicht as efficiently as possible and also to enable the application developer to make efficient use of the machine. Based on these considerations, we chose to use C++ directly, with no higher language abstraction built on top of it. However, we try to define a high level of abstraction for Neonlicht, and focus on the object-oriented features of C++. By this we hope to reduce the amount of code writing and simultaneously focus on the audio processing code.

Two effects can be achieved by the (initial) restriction onto the Raspberry Pi. First, the systems can be inexpensively purchased in larger numbers and then passed on to students (either for a limited time or permanently). These *identical* sets are then available in the classroom, which greatly facilitates the explanations of the lecturer. Secondly, it is possible to use the small computer in virtually any location without major problems. Thus, it will become much simpler to present the projects of the students in various public contexts.

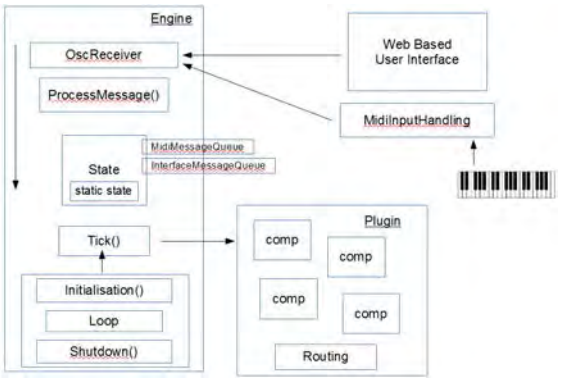


Figure 2. The system architecture of the Neonlicht engine.

4. NEONLICHT: BUILDING THE SYNTHESIZER ENGINE

When planning Neonlicht we assumed that the communication between the operating system and the audio hardware could be realized with existing software libraries.

For audio input and output we chose *RTAudio* and correspondingly *RTMidi* for communication with and control of the Midi hardware (see [2]). These libraries are very mature and compile and run fine on the Raspberry. They are thus a very stable foundation for the abstraction that should be made by Neonlicht.

Besides using RTAudio/RTMidi, Neonlicht also uses a few parts of the STK. This takeover of functionality includes four standard filter classes that are implemented in the STK. As these filters are not a central part of the intended functionality of the system, we simply used what has already been implemented in a very reliable way. In

addition, this takeover demonstrates how to integrate functions from other libraries into the Neonlight architecture.

For the implementation of the network communication, the OSC library oscpack (see [6]) was used. The loading and storing of configuration data is done with config4star (see [7]). Unfortunately, these two libraries are apparently no longer being developed. For this reason the complete source code of the libraries has been integrated into the distribution of Neonlicht as a 3rd party software. This ensures, that at least the current state of the libraries will be available on a near term basis. The licenses of the two libraries allow for such redistribution.

4.1 Audio programming with Neonlicht

The system architecture of Neonlicht separates

- the code for the audio processing and
- the code for the control via GUI or MIDI

from the actual engine (see Figure 2).

The application developer has to implement a plugin, a so called *Sound Unit*. For doing this a simple interface with four methods has been defined. When configuring a Neonlicht application the sound unit must be passed on to the engine. Neonlicht controls the sound unit and forwards the generated samples on to the audio hardware.

The implementation of the sound units follows the well established approach of unit generators (see [8]). Neonlicht itself currently provides about 60 unit generators. From these, more complex sound units are constructed, which are then also unit generators and can be used to recursively build up more complex units. For students who have mastered the concept of unit generators, the development of an audio system in Neonlicht turns out to be very comprehensible.

By this approach the amount of application code is reduced greatly. In order to implement a simple subtractive synthesizer called *Workshop-16*, that simulates the functionality of the provided Moog systems, only 34 lines of audio code had to be written. The granularity of the used unit generators corresponds to that of comparable current systems. The code for the system control is a little longer, but basically consists of a mapping of command strings to system parameters.

In the end the user has access to a synthesizer, that is fully playable with a standard Midi Controller, being able to change sound parameters in real time. The software runs on a Raspberry 3, causing rarely more the 10% system load. This number is taken on the development system, where a full X-Server is running, taking away considerable amounts of processing time.

Reliable statements about the exact performance of the engine can not yet be made, although there are measurements on the runtime behavior of some of the unit generators. The audio sampling rate of 44.1 KHz corresponds to a maximum tick()-time of 22.675 microseconds. This interval must not be exceeded by the audio processing, since otherwise drop outs in the audio stream will occur. As can be taken from the data in table 1 the average processing times of the unit generators listed are significantly below this threshold.

unit generator	*time in μs*
noise1→tick();	0.121
saw1→tick()	0.0314
mixer1→tick()	0.447
square1→tick()	0.0281
onelpf→tick()	0.0389
eg1→tick()	0.053
cosine1→tick()	0.100
phasor1→tick()	0.052
Workshop-16→tick()	1.472

Table 1. Time measurements on a Raspberry 3 of a single tick() function call in a subset of the implemented unit generators. The maximum time span must be less than 22.675 μs, otherwise audio drop outs will occur.

According to these measurements, the system has sufficient performance reserves to implement more complex real-time audio systems. In theory up to 14 Workshop-16 systems can run in parallel on the Raspberry 3!

The most complex code part when developing a synthesizer with Neonlicht turns out to be the implementation of the graphical user interface. Of course the user can always do without implementing a GUI, as it is not in any way required by the Neonlicht architecture.

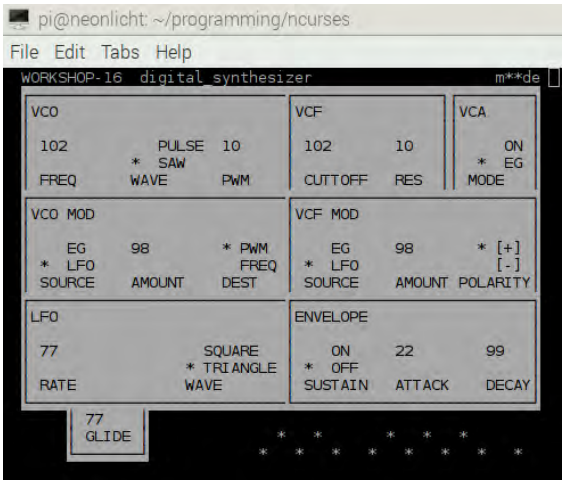


Figure 3. The ncurses interface of the Workshop-16 synthesizer. The GUI of a sound unit developed for Neonlicht could be implemented in almost any technology. For this presentation we have chosen to show a very basic, (and very retro) text based version running in a standard terminal window.

4.2 Integration of a GUI

Figure 3 shows the exemplary implementation a graphical user interface for a more complex sound unit. In this case, it is the simulation of a synthesizer with simple subtractive synthesis. The synthesizer uses standard components

(VCO, VCF, ADSR, LFO etc.) in a standard routing configuration. Accordingly, the user interface has been designed: the central parameters of each component can be controlled and the signal routing can be adjusted within certain limits.

As you can see the presented GUI is realized using a very reduced, text oriented design. By this, we wanted to point out that the user interface can be realized in any technology, including text based UIs, web based UIs or application based UIs. The developer is free to choose from any technology, as long as it provides the ability to communicate with Neonlicht via OSC.

5. USING NEONLICHT: FIRST IMPRESSIONS

The next iteration of the audio programming course will be held in the up coming winter semester. In order to get a first impression of whether Neonlicht can be considered a didactically meaningful tool, we introduced Neonlicht to the participants of last year's course. In a 1-day seminar they were asked to try to port their audio applications to Neonlicht. This is admittedly a slightly different situation than a standard course, as the students have significantly higher knowledge in audio programming. In fact, the students responded very positively to the engine. No one has had any experience in programming in C++, still, with the support of the teachers, they managed to successfully reimplement large parts of their audio applications. They were particularly inspired by the fact that, at the end of the day, a single executable program existed, that performed the complete audio processing.

6. CONCLUSIONS

In this paper we have provided an overview of the synthesizer engine Neonlicht. The system is oriented towards the efficient development of audio applications on the Raspberry 3 and offers a high level of abstraction.

In order to have a sufficient performance, Neonlicht omitted the development of a virtual machine with an associated interpreter for a domain-specific language. Instead, the abstraction provided by Neonlicht should facilitate the immediate implementation of audio processing units in C++.

Overall, the illustrated approach of Neonlicht integrates well with a didactic concept of a stepwise teaching method with practical exercises, where existing tools are used to teach basic concepts of audio programming.

By the subsequent implementation of these processes in Neonlicht students are able to transfer the learned concepts into powerful audio applications.

Acknowledgments

This work is to a large extent the result of my sabatical research semester at Linneaus University, Växjö, Sweden. I would like to thank the colleagues of the computer science department of technical faculty for their support during this very productive stay.

7. REFERENCES

[1] M. S. Puckette, "Pure data," in *Proceedings, International Computer Music Conference*, San Francisco, 1996, pp. 224–227.

[2] P. R. C. Gary P. Scavone, "Rtmidi, rtaudio, and a synthesis toolkit (stk) update," in *Proceedings of the 2005 International Computer Music Conference*, Barcelona, Spain, 2005.

[3] G. Wang, P. R. Cook, and S. Salazar, "Chuck: A strongly-timed computer music language," *Computer Music Journal*, vol. 4, no. 39, pp. 10–29, 2015.

[4] S. Wilson, D. Cottle, and N. Collins, *The SuperCollider Book*. MA: The MIT Press, 2011.

[5] R. Foundation", "Raspberry 3 online," in *https://www.raspberrypi.org/products/raspberry-pi-3-model-b/*, 2016.

[6] J. Kleimola and P. McGlynn, "Improving the efficiency of open sound control with compressed address strings," in *Sound and Music Computing Conference (SMC-2011)*, Padova, Italia, 2011, pp. 479–485.

[7] C. McHale., "Config4star online," in *http://www.config4star.org/*, 2016.

[8] M. Mathews, "An acoustical compiler for musical and psychological stimuli," Bell Telephone System Technical Journal, Tech. Rep., 1961.

ZIRKONIUM, SPATDIF, AND MEDIAARTBASE.DE; AN ARCHIVING STRATEGY FOR SPATIAL MUSIC AT ZKM

Chikashi Miyama
ZKM Karlsruhe
miyama@zkm.de

Götz Dipper
ZKM Karlsruhe
dipper@zkm.de

Robert Krämer
ZKM Karlsruhe
kraemer@zkm.de

Jan C. Schacher
ICST Zürich
jan.schacher@zhdk.ch

ABSTRACT

ZKM | Institute for Music and Acoustics has been contributing to the production and realization of spatial music for more than 20 years. This paper introduces how the institute archives the spatial compositions, maximizing the universality, reusability, and accesibility for performances and research in the future by combining three key elements: Zirkonium, SpatDIF, and mediaartbase.de.

1. INTRODUCTION

1.1 IMA

The Institute for Music and Acoustics (IMA) at the Center for Art and Media (ZKM) in Karlsruhe is dedicated to electroacoustic music. One of the main focuses of the IMA is the creation of new electroacoustic compositions as well as their presentation. For this purpose the IMA conducts a guest artist program, where we invite on average about 30 composers per year to work in one of our ateliers. Such a residency normally lasts one to three months and the result is typically a new electroacoustic composition. In the course of more than 25 years of the institute's activity since 1989, several hundred pieces have been composed. Thus, it is obvious that a strong archiving strategy is very important for us, which led us to the *mediaartbase.de* project described in section 3.3.

The pieces composed at IMA cover various kinds of electroacoustic and computer music, including live electronics and fixed-media pieces. Already from the very beginning of IMA in 1989, most of the new compositions were multichannel pieces, often for the widely-used quadraphonic or 8-channel circle of loudspeakers. This fact is reflected by our atelier infrastructure – most of our ateliers are equipped with four or eight loudspeakers.

In 2006, we introduced the *Klangdom* ("Sound Dome"), which can be considered as a quite natural extension and successor of the quadraphonic and 8-channel circles. It is a speaker configuration in the form of a large hemisphere surrounding the audience [1]. Our main Klangdom is installed in the ZKM_Cube, which is the IMA's main concert space. In order to facilitate the control of the Klangdom we developed the software *Zirkonium*. It allows the composer to spatialize sounds by placing them onto the hemisphere of speakers and by moving them arbitrarily between speakers. Zirkonium will be discussed in more detail in section 3.1. Today, more and more composers decide to create pieces for the Klangdom. Therefore, it is inevitable to ponder the archiving strategies of this kind of spatialized music. This paper focuses on the fixed-media pieces composed for the Klangdom and their archiving issues. Some parts of the discussed problems may apply to a broader range of electroacoustic compositions.

1.2 Objectives of archiving at IMA

Concerning the target audience of our archive at the IMA, we may distinguish three different scenarios.

- Preservation, long-term availability: the archived compositions should be preserved as long as possible.
- Application, short-term availability: The archived compositions should be available in a format which is ready to be played by current spatial audio systems. In other words, a composition can be copied from our archive and can immediately be played on our concert system, without the need of any cumbersome and error-prone format conversion. This is quite important for our everyday commodities within the IMA.
- Accessibility: the archived compositions should be accessible for the general public as well as for researchers and scholars who are interested in them.

2. ISSUES

2.1 Object-based vs. channel-based audio

2.1.1 Number of speakers

The mentioned transition from quadraphonic or 8-channel circles to the Klangdom and Zirkonium went along with the transition from so-called channel-based audio to object-based audio. In channel-based audio all spatial information is contained in the audio data, while in object-based audio the audio data is accompanied by separate meta data which defines the spatial behavior of the audio [2]. A consequence of this difference is, that in channel-based systems the speaker layout is predetermined and cannot be changed, while in object-based systems the number of speaker channels is flexible. The actual number and position of speakers in object-based systems is known by the software which renders the movements of the audio at run-time. Consequently, the number of speakers is

not a relevant information for the archiving of Klangdom pieces, while it is obviously an essential information for the archiving of channel-based pieces.

2.1.2 Scores

The meta data which defines the spatial behavior of the audio data can be received by the renderer in different ways. In Zirkonium there are basically two ways. One way is via the OSC protocol, which is frequently utilized for live applications. The other way is via a spatialization score, which is the preferred way for fixed-media pieces. In order to create a score, the composer can use Zirkonium's *Trajectory editor*, where he can graphically draw the paths along which the audio should travel. Until recently Zirkonium used its own XML format for the score. The newest version of Zirkonium, version 3.2, offers a functionality to export more universal *SpatDIF* version 0.4-compliant XML. SpatDIF is discussed in more detail in section 3.2.

2.1.3 Documentation

A significant advantage of spatial score is that the score represents a documentation of the audio movements. This can be extremely helpful for researchers, when it comes to the analysis of the spatial characteristics of a composition. And since researchers are among the main target users of our archive, object-based pieces are in this sense better suited for the archive than channel-based pieces. Obviously, the score of spatial notation is beneficial not only for the researcher, but also for the composer himself, because it makes it easier to revisit and revise old pieces or reuse materials for new pieces.

There are different degrees of abstraction as to how a movement can be described in the score. The higher the level of abstraction, the better it accommodates the analysis of the spatial characteristics of the composition. In a high level of abstraction, an instruction might be similar to the following: "audio object xy should be cycling clockwise for xy seconds with xy angular velocity." In a lower level of abstraction the same movement might be described by a series of discrete breakpoints.

2.1.4 Advantages of channel-based audio

There are also advantages of channel-based audio compared with object-based audio. The main advantage is its simplicity. It requires neither a score nor an application, which would interpret and render the score. The knowledge of the speaker position is sufficient in order to play the piece using any DAW software, any computer platform or even just a tape machine. For Zirkonium pieces, on the contrary, the performer still needs the Zirkonium software, a Mac computer with a compatible operating system, a compatible audio interface, etc. In this sense object-based pieces are less suited for long-term archiving – which on the other hand is one of our main archiving objectives. In order to overcome this disadvantage, it is necessary to establish a standard for the spatialization score, which is the aim of SpatDIF. Zirkonium changed its score format to be SpatDIF compliant in order to support this endeavor.

2.1.5 Exporting Zirkonium pieces

Object-based pieces can be converted to channel-based pieces by simply recording the audio signal which is being sent to the speakers. In this way, we could retrieve a channel-based version of a Klangdom piece which matches exactly the speaker layout of the Klangdom in the ZKM_Cube. One might call this a "bounced" version of the Zirkonium piece. This channel-based version would then have all the advantages and disadvantages of any other channel-based piece. We use this method mainly with Klangdom pieces which have been created with other software than Zirkonium, because in these cases we do not have influence on the software development and thus we cannot be sure about the future availability of the software.

Strictly speaking a channel-based version can only be played on a speaker setup which matches exactly the original setup. Nonetheless there is a workaround for Klangdom setups which differ in their number of speakers: The original speakers are placed as virtual speakers onto the actual target Klangdom. It might cause a loss of spatial resolution either compared to the original piece (if the original Klangdom comprises more speakers than the actual Klangdom) or compared to the potential of the actual Klangdom layout (if the actual Klangdom comprises more speakers than the original one). Nevertheless it seems to be a reasonable way to treat Klangdom pieces, if there is no object-based, but only a channel-based version available.

2.2 Audio file format

A detailed discussion about the audio file format appropriate for archiving would exceed the scope of this paper. We want to mention only a few issues which we faced in the context of Klangdom pieces.

The main issue was that different major versions of Zirkonium needed different audio file formats. The first, "classic" Zirkonium, released in 2006, was able to playback only a limited numbers of mono sound files, because it produced drop-outs during the file reading process. The solution was to pack ideally all audio sources in one monolithic interleaved file, containing up to 32 channels, which used to be the maximum number of channels "classical" Zirkonium could afford to process. A piece with 32 source channels, 48 kHz sampling rate, 24 bit sample depth and a duration of 20 minutes would require a file size of more than 5 GB. This was beyond the limit of 4 GB, set by the commonly used WAV or AIFF file formats. Therefore, the recommended file format for classic Zirkonium pieces was the Core Audio Format (CAF) [3], released by Apple in 2005.

Zirkonium version 2, released in 2014, replaced the CoreAudio-based audio engine of "classical" Zirkonium by Max/MSP and employs sfplay˜objects for playback. However, sfplay˜is unable to play CAF files. The recommended file format for Zirkonium version 2 pieces was 8-channel WAV format, using several audio files, if more than eight source channels were needed. The conclusion drawn from these audio file format issues is, that for short-

term availability the file formats in the archive might have to be changed from time to time.

Concerning long-term availability, we tend to consider a collection of mono files as the most secure way of archiving, because in that way the channel assignment remains stable even if parts of a file get corrupted or lost. Therefore, we have to archive a piece in different versions for short-term and long-term availability.

Another issue is the sampling rate. We try to avoid changing the sampling rate during a concert. Most concerts are run in 48 kHz, but there are also concerts in 44.1 kHz. Given our archiving objective of short-time availability as described above, this means that we should keep two versions of each Klangdom piece in the archive, one for each sampling rate. The problem arises that easily large amounts of data accumulate and have to be held available. This is especially true if variants for different major versions of Zirkonium should be available as well as bounces for different Klangdom setups.

2.3 ZirkOSC

There is a special group of Klangdom pieces in which the spatialization data is not stored in a Zirkonium file, but in a DAW session file as automation data. In these pieces, the Zirkonium-Renderer is remote-controlled via OSC messages sent from the DAW plugin *ZirkOSC*. ZirkOSC is a separate software being developed since 2012 by the Groupe de Recherche en Immersion Spatiale (G.R.I.S.) at Université de Montréal under the direction of Robert Normandeau [4]. A similar approach is taken by the ToscA plugin developed by IRCAM in 2014 to control IRCAM's Spat software [5]. The advantage of ZirkOSC or ToscA is their seamless integration into the DAW workflow. The disadvantage is that the automation data is tightly bound to the actual DAW software and it is difficult, if not impossible, for the archive to provide long-term availability.

There are two solutions for this problem. One solution is to bounce the piece as described in section 2.1.5, which will result in a channel-based version. The other solution is to record all the automation data in Zirkonium version 2 once the piece is completely finished. This can be done automatically and yields an object-based, stand-alone, native Zirkonium version of the piece. We plan to implement this feature in Zirkonium version 3 in the future.

2.4 Binaural Versions

Binaural recordings of Klangdom pieces are maybe only of peripheral interest concerning this paper's topic. Nevertheless we want to mention them as one form in which Klangdom pieces can be distributed. They are of main importance for people without immediate access to a Klangdom environment. Since many researchers and the general public can be included in this group of people, the topic is indeed relevant for our archiving objectives.

3. KEY ELEMENTS OF ARCHIVING AT ZKM | IMA

3.1 Zirkonium

3.1.1 Overview of the software

Zirkonium is a software suite for spatial composition and performance. The development of the software was begun in 2004 and it has been continuously refined until today. In November 2015 the newest version of Zirkonium, version 3.0 was released [6], and version 3.1 is planned to be released in June 2016.

Zirkonium ver. 3.1 consists of three applications. *Trajectory editor*, and two utility applications, *Speaker setup* and *ZirkPad*.

The Trajectory editor (Fig.1) is a standalone Mac OSX software that allows users to compose trajectories of multiple sound objects in time and render them to a dome-like speaker setup consisting of a maximum of 64 speakers. The software features an OpenGL-based superior graphical user interface for trajectory creation, DAW-like spatial event handling tools, and customizable core rendering algorithms.

An accompanying utility software, Speaker Setup, provides users with an OpenGL-based GUI that enables them to define speaker arrangements in a 3D space intutively. The configuration created by the Speaker Setup application can be stored in either Zirkonium-XML format or SpatDIF-XML format. The Trajectory editor then employs these XML file for initializing its spatial rendering algorithms and visualizing trajectories.

Zirkonium also offers live-performance capabilities. The position of each sound object can be controled by sending OSC messages from an accompanying iPad application, ZirkPad or any other OSC-compatible software. This feature allows users to utilize Zirkonium in a live situation as well.

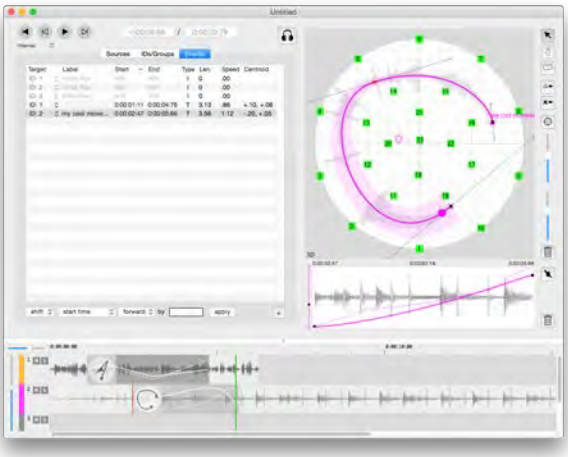


Figure 1. Zirkonium ver. 3.1 Main Window

3.1.2 The data structure of Zirkonium event

In Zirkonium, a spatial composition project consists of multiple spatial *events*. These events are displayed in the

manner of a typical DAW software in a timeline view at the bottom of main window (Fig. 1). An event has several attributes, such as target, label, start time, end time, sound path, motion path, span automation, and volume automation (Fig. 2).

Among these attributes, *target* means a sound object or a group of multiple sound objects that is associated to the event. Start and end time define the time frame of the event. The actual trajectory (i.e. the movement of an individual sound object, or a group of sound objects) is described by the combination of two *paths*: Sound path and Motion path. The Sound path defines the geometrical route of a sound object in the listening space, and the motion path defines how a sound object moves in time between the start point (marked with a triangle symbol) and the end point (marked with a cross symbol) of the defined sound path (Fig. 2).

A sound path is stored in the form of a multi segment cubic bezier curve, and the motion path is stored as a multi segment exponential curve. These data for trajectory are rasterized and stored both in the RAM and VRAM each time the user modifies or load a pre-existing file to the Trajectory editor. The rasterized data will be employed for the playback, visualization as well as SpatDIF layer 5 export, described later.

The span automation controls the spread of a sound object (i.e. the diffusion of sound in the listening space) over the time frame of an event, and the volume automation manipulates directly the level of the sound object. These two automations are also described with multi-segement exponential curves.

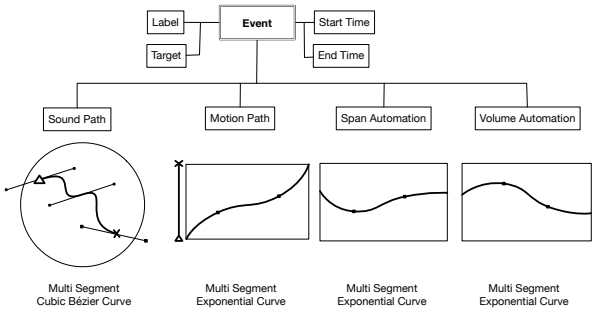


Figure 2. The data structure of an Event in Zirkonium

As briefly disscussed in section 2.1.2, until the release of the Zirkonium version 3.0, a collection of spatial events (i.e. a score of spatial music) was stored only in the file format generated by Apple's core data framework. The core data file can only be loaded by the Zirkonium trajectory editor and does not offer any compatibility to other software. However, Zirkonium ver. 3.0 offers a function to export spatial events using SpatDIF, but the exported SpatDIF data was only available in the form of rasterized descrete data of spatial movements; it was impossible to retrieve the data of Sound path or Span automation from the SpatDIF file.

Zirkonium ver. 3.2, which is planned to be released in September 2016, will be able to store these paths as vectorized data using the most up-to-date version of SpatDIF specification and its C++ library, which will be described in detail in the following section.

3.2 SpatDIF

The Spatial Sound Description Interchange Format SpatDIF is a structured, high-level syntax describing the essential elements of spatial sound scenes that are necessary for their creation and performance. It proposes a simple and extensible format as well as best-practice examples for storing and transmitting information about spatial sound scenes. It is a syntax rather than a programming interface or file-format and can be represented in any of the current or future structured mark-up languages or messaging systems [7].

The human-readable descriptors are structured in a hierarchical fashion, divided into a core category, essential functionalities, and optional extensions. Based on a layered workflow for spatial audio content, elements of such diverse nature as source positions, media playback and patching information from abstract scene to a specific speakers layout are present. This model organizes spatial audio infrastructure and elements in the following layers:

6. authoring – generates scene control data (vectorized)
5. scene description – generates rendering instructions (rasterized)
4. audio stream encoding – actual rendering
3. audio stream decoding – rendering in two-step processes such as Ambisonics
2. hardware abstraction – system audio drivers
1. physical devices – soundcards and speakers

Two principal use-cases can be distinguished:
The first scenario is focusing on storing spatial audio scene descriptions for future playback. This file-based representation contains all relevant scene information within one file, divided into a meta-section with preparatory information and a temporal section with containing the unfolding scene. In addition to the SpatDIF scene description file the actual audio content needs to be stored and maintained alongside it.

The second scenario deals with streamed audio content and scene description information in real-time and quasi real-time. In this network-based representation the scene information is delivered piecewise in time as the scene unfolds and is not guaranteed to provide all necessary setup information in advance. In this scenario contradictory information may arrive that supersedes earlier instructions, and audio content may be streamed alongside the scene information or may be present as sound-file accessible to the rendering software.

The principle of interoperability means that these pieces may originate and be processed by different tools at separate times and places, including future system whose capabilities are yet unknown. The exact nature and technical implementation of these processes should not have to be known or determined at the outset.

A central concept in SpatDIF is the separation of processes of *authoring* and *rendering* of spatial sound scenes or musical pieces such as those in the Zirkonium catalog.

The presence of the *core scene description* (layer 5 information) is essential for generating a *rendering*, i.e., for (re-)producing the piece as musical or sound performance. This representation carries all necessary scene information in its simplest possible form that does not demand complex pre-processing of information in order to obtain actionable rendering parameters. With the addition of simple interpolations, the scene entities' attributes, such as positions or volume changes are described explicitly in sufficient temporal resolution to enable smooth rendering. Every SpatDIF compliant data-set therefore *must* contain such a core scene representation. Even though this essential information depicts a complete scene, it lacks information about the processes and models that led to its creation, from where the author's ideas and intentions could be deduced, recreated or even adjusted.

Therefore version 0.4 of SpatDIF [8] adds descriptors that represent aspects of *authoring* processes (layer 6 information). These processes are typically carried out in an editor such as the Zirkonium Trajectory editor (see Fig. 1) or media programming environments such as MaxMSP, PureData etc. but are also present as automation-data in a DAW's project. This concerns typically the placement and animation of sound sources and describes the developments of temporal, spatial and volume attributes of the entities in the scene (see Fig. 2). Even though this higher-level authoring information is sufficient to recreate a spatial audio piece within the software it was created with, the goal of interoperability and reproduction in future software tools of unknown capability *demands* that the abstract representation, be converted to the simpler core descriptions, i.e., converting the 'vectorized' to 'rasterized' information.

With the addition of authoring descriptors, the SpatDIF data-set contains redundant information in two representations at different levels of abstraction (layer 5 *and* 6).

In such a case, a *rendering* process disregards the authoring information, whereas an *authoring* or editing process that modifies the scene's animation processes supersedes the simpler scene rendering information. In order to maintain the two representations synchronous, when storing the scene, the modifications of the 'blueprint' of the scene in the authoring layer, i.e., of the models and functions that describe the evolution of scene, are always propagated down to the simpler representation. Consequently, the existing rendering instructions are potentially overwritten.

3.3 mediaartbase.de

While Zirkonium offers an interface to the Klangdom and a tool for the spatialization of a composition, SpatDIF guarantees the interoperability and exchangability of the composition. A solution for the storage and systematic description of such media art has long been unavailable. mediaartbase.de seeks to close the gap between the compositional process, the finished production and the presentation of electroacoustic music.

In 2008, four major institutions in the field of media art started the project mediaartbase.de (Fig. 3) [9]. The Institute of Music and Acoustics (IMA) at the ZKM Karlsruhe,

Figure 3. mediaartbase.de website

the European Media Art Festival Osnabrück, the documenta Archive Kassel and the "Kasseler Dokumentarfilm- und Videofest" came in order to create a platform that suited the need for the preservation and systematic documentation of the miscellaneous works, which have been curated or produced at these institutions. The necessity for such an endeavour was acknowledged by the "Kulturstiftung des Bundes" and therefore financially supported [10]. It was a crucial goal for the design and concept of a prospective database to combine the long-time archiving approach with the availability of the archival objects for the general public. This led to an overall design which separates the (publicly available) presentation sphere from the (long-time) archiving area. Although the appearance as well as the basic structure of the collections from the cooperating partners ("communities") on mediaartbase.de are similar, the structural subdivisions ("collections") of each institution may vary. [1] Therefore the following section focuses only on the strategies, concepts and ideas currently embedded in the archiving process at the IMA. The final report of the project offers further details including the cooperating partners [11].

The adequate description of the work produced at IMA is fundamental for its presentation and archiving. On mediaartbase.de basic information about each production is stored using the Dublin Core Metadata Element Set, which is used by the other institutions in a complementary fashion to extend their own data set, hence guaranteeing a coherent and uniform description for the key information of a piece.

Besides Dublin Core, mediaartbase.de relies on three different metadata registries, which employ the namespace of MARC [12], RDA [13] and MODS [14], enabling the different institutions to enhance the existing metadata design

[1] E.g. the Kasseler Dokumentarfilm- und Videofest decided to create a singular "collection" for every Festival while the IMA chose to structure their collections more broadly ("Audiovisual Productions", "External Musicproductions", "Artists", "Publications" and "Presentations").

Descriptors	Value
dc.title	Title
dc.creator	Author
dc.type	Type [Sound/Video]
dc.type.form	Genre
dc.description	Description
dc.placeoforigin	Country
dc.format.extent	Duration
dc.relation	Catch-all for references to other related items
loc.producer	Producer

Table 1. Shared Descriptors used by all institutions in mediaartbase.de (DC and MARC).

and model,while also offering a shared space for the development and management of the necessary descriptors.

The IMA currently manages six different collections in which productions are arranged by type (audiovisual productions, music productions), while also containing external music productions, publications, events and information about artists. Relationships between different entries in those collections can be established by using the relevant descriptors, allowing the rich and comprehensive history of a single production to be stored as well. [2] This has been made possible by a handle system, that assigns each entry a persistent identifier.

The underlying open source database DSpace not only provides this handle system, but also allows for an easily adjustable rights management [15]. Currently there are two basic options to access the online platform: as registered or unregistered user. The archived material can therefore be displayed either in an unrestricted or restricted manner, based on which rights the artist granted the IMA. Different usergroups can be created and different access permissions may be set up. An IP-based access, allowing the unrestricted display and playback of the archived material within the rooms of the ZKM has been installed in 2015 and offers visitors and employees of the media museum an insight of the various music productions. Due to the separation of archiving and presentation only the metadata and more compressed formats of the stereo versions are available online. The archived originals (multichannel versions, Zirkonium files, uncompressed stereo versions) of the productions are stored locally and held available for both research and performance. However, the technical description (sample rate, bit depth etc.) of the originals are available online and can be accessed, when logged in. The link between the online items and their originals is established through a signature. mediaartbase.de is therefore meant to be a first stepping stone for exploring media art as well as an approach of a systematic overview of the archived material.

The idea of exploring media art systematically is mirrored in the frontend of mediaartbase.de as well. The user is able to browse all collections of a community by categories (“Date”, “Authors”, “Titles” etc.) or to use the search option, that allows for a more refined access to the available database objects. Therefore it becomes a tool for a curious public audience.

On the other hand mediaartbase.de offers access statistics, adaptable submission forms and a supervised system for the submission process that simplifies the workflow and the controlling of the published items. Therefore, mediaartbase.de helps in structuring and systematically recording the archived material as well.

4. WORKFLOW

Even though nowadays more and more guest artists are using Zirkonium for the productions at IMA, the workflow of the archiving process is still determined by the original idea of recording, storing and describing old fixed media pieces from DAT and HI8 (DTRS) to HDD, thereby saving them from the nearly unavoidable degradation of the medium. But the compositional and technical process in the production of electronic music has moved on. It seems that the problem of different versions of a musical piece has grown with the possibility of the composer making changes very simply using only his personal computer rather than a whole studio. This also led to a more reluctant position of composers towards the archiving of their piece, since this is often perceived as a final commitment on behalf of the composer. However, many composers are still interested to see their work archived and presented in an appropriate manner.

Another important change regarding the archived work is that while fixed media pieces from DAT and DTRS are bounced to several mono-channel files resulting in the finished production of a composer, the Zirkonium pieces only consist of the input files together with the respective Zirkonium-file, describing the spatialization process. Even though, as mentioned above, it is possible to bounce the finished work to several mono-channel files in Zirkonium as well, this would lead to unnecessary redundancies. Since the Zirkonium-project of a composer can be understood as a sort of musical score, it may contain valuable information for researchers and therefore should be stored. It is important to remember, that the pieces created with Zirkonium are not bound to the maximum of loudspeakers in the described Klangdom. Therefore, a bounced version can only be a momentary record of the work of a composer. Furthermore, the newly added descriptors in SpatDIF 0.4 maintain the abstract representation of musical score, while increasing the compatibility of data. It is a clear advantage of SpatDIF compared to other specifications such as ADM [16]. Though most of the major institutes of electro-acoustic music employ the channel-based archiving approach, we adopted the new object-based approach because it greatly contributes to our three objectives: preservation, application, and accessibility.

Based on the experience with the first version of Zirkonium, the second version introduced major changes, which made the composition and the performance of the produced work more comfortable. Currently, the backward

[2] Besides the obvious link between a production and its presentation on a publication or within a concert (dc.relation.ispartof/dc.relation.haspart), further relationships can be modeled. E.g. different version of a piece (dc.relation.isversionof/dc.relation.hasversion).

compatibility of Zirkonium ver. 3.2 is being implemented. IMA has to retain and maintain the first version until a sufficient number of tests are performed. This generates additional work in the documentation process of spatial music, but also affected the performance of these works, since the performance hardware in the Klangdom needed to support both versions as well as the barrier-free switching between them. Having learned from this situation, the decision to rely on a file-format that doesn't depend on a software and its versioning history proved to be almost unavoidable.

Since the early Zirkonium file format is based on xml (zrkpxml) the retrieval of basic information is possible, but, due to the fact that binary information (e. g. color information) was stored as well, the human readability suffered from this. The ensured design and maintaining of the specifications of a file format for performance and archiving purposes require to address exactly these points. Currently, archiving at IMA comprises multiple steps. After the composition process, a guest artist at ZKM usually hands his produced work over to the IMA archive. Based on a questionaire given to the artist, the relevant information about the piece and a work abstract is added to mediaartbase.de. A stereo version provided by the composer is also used for the presentation of the piece as well as additional files such as image material or scores. Within an archive-contract the composer has the possibility to exactly determine how accessible his work should be and whether a full stereo version is available or only a short sample of it. It is our aim to make this process as easy and unbureaucratic as possible. Besides other measures, using SpatDIF as a source for information can help making the information retrieval process more transparent and less complicated. A file format which contains the important information in one clearly documented way is the next step in order to reach this goal and to ensure the further cooperation of the composers.

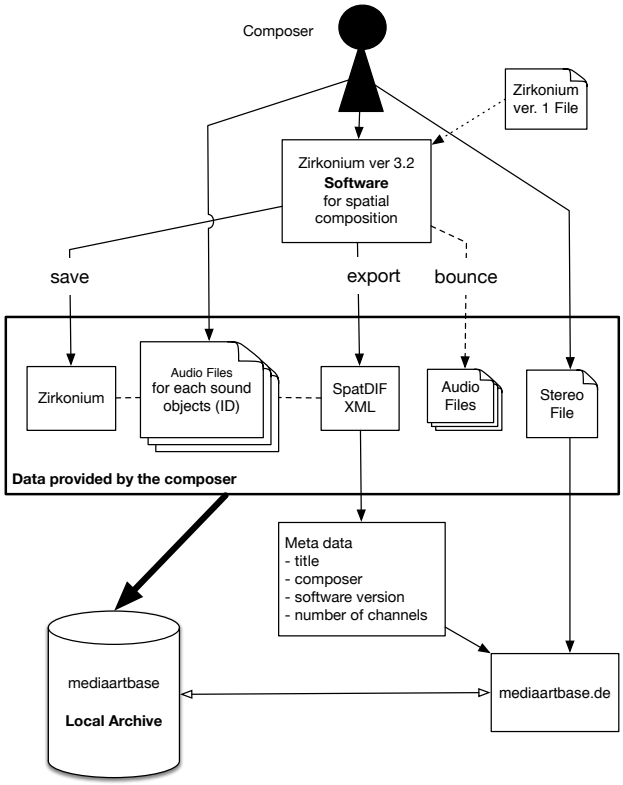


Figure 4. The workflow of archiving at ZKM

5. CONCLUSIONS

Preservation, application, and accessibility are three primary objectives of spatial music archiving at IMA. In order to achieve these objectives, the IMA deploys three key elements: Zirkonium, SpatDIF, and mediaaartbase.de.

The newest version of Zirkonium contributes to reaching these objectives by making spatial compositions available in a wide range of formats; it allows users to export a piece to simple channel-based audio files together with a more flexible SpatDIF 0.4 compliant spatial score.

The use of SpatDIF for storing Zirkonium pieces in the mediaartbase.de represents one of the central use-cases this standard was developed for. At the same time, it provides an ideal test-case for validating that the newly defined authoring layer descriptors are capable of properly representing the high-level information of a real-life trajectory editor.

It should be evident that a file format, which is future-proof, human-readable and independent from specific platforms or applications is imperative for the archiving process at IMA. Since the aim of mediaartbase.de is not only to store and preserve the archived media, but also to make it available and accessible for research and performance, a format which considers these different and sometimes conflicting demands is hard to find. The SpatDIF format can be integrated easily within our archiving workflow and therefore perfectly complements and supports our endeavor for a coherent, cohesive, and systematic production archive.

Acknowledgments

We would like to thank Ludger Brümmer, director of the IMA, Robert Normandeau, and Markus Noisternig, who provided further insight and expertise that greatly supported us.

6. REFERENCES

[1] C. Ramakrishnan, J. Goßmann, and L. Brümmer, "The zkm klangdom," in *Proceedings of NIME*, Paris, 2006, pp. 140–143.

[2] F. Melchior, U. Michaelis, and R. Steffens, "Spatial mastering - a new concept for spatial sound design in object-based audio scenes," in *ICMC*, Huddersfield, 2011, pp. 5–8.

[3] A. Inc. Caf file overview. [Online]. Available: https://developer.apple.com/library/mac/documentation/MusicAudio/Reference/CAFSpec/CAF_overview/CAF_overview.html

[4] Zirkosc3. [Online]. Available: https://sourceforge.net/projects/zirkosc3/

[5] T. Carpentier, "Tosca: An osc communication plugin for object-oriented spatialization authoring," in *Proceedings of ICMC*, North Texas, 2015, pp. 368–371.

[6] C. Miyama, G. Dipper, and L. Brümmer, "Zirkonium mk iii - a toolkit for spatial composition," *Journal of the Japanese Society for Sonic Arts*, vol. 7, no. 3, pp. 54–59, 2015.

[7] N. Peters, T. Lossius, and J. C. Schacher, "The Spatial Sound Description Interchange Format: Principles, Specification, and Examples," *Computer Music Journal*, vol. 37, no. 1, pp. 11–22, 2013.

[8] N. Peters, J. C. Schacher, T. Lossius, and C. Miyama. (2010-2016) Specification of the spatial sound description interchange format (SpatDIF) v. 0.4. [Online]. Available: http://spatdif.org/specifications.html

[9] mediaartbase.de. [Online]. Available: http://mediaartbase.de

[10] Kulturstiftung des bundes. [Online]. Available: http://www.kulturstiftung-des-bundes.de

[11] L. Brümmer. Abschlussbericht mediaartbase.de. [Online]. Available: http://193.175.110.9/hornemann/german/epubl_txt/2013_KUR_Projekt_Bruemmer.pdf

[12] Marc. [Online]. Available: http://www.loc.gov/marc/

[13] Rda. [Online]. Available: http://www.rda-rsc.org/

[14] Mods. [Online]. Available: http://www.loc.gov/standards/mods/

[15] Dspace. [Online]. Available: http://www.dspace.org

[16] Audio definition model. [Online]. Available: https://tech.ebu.ch/docs/tech/tech3364.pdf

FONASKEIN: AN INTERACTIVE SOFTWARE APPLICATION FOR THE PRACTICE OF THE SINGING VOICE

Fotios Moschos
University of Athens,
Department of Pedagogy,
Athens, Greece
fotmos@windowslive.com

Anastasia Georgaki
University of Athens,
Department of Music Studies,
Athens, Greece
georgaki@music.uoa.gr

Georgios Kouroupetroglou
University of Athens,
Department of Informatics and Telecomunications
Athens, Greece
koupe@di.uoa.gr

ABSTRACT

A number of software applications for the practice of the singing voice have been introduced in the last decades, but all of them are limited to equal tempered scales. In this work, we present the design and development of FONASKEIN, a novel modular interactive software application for the practice of singing voice in real time and with visual feedback for both equal and non-equal tempered scales. Details of the Graphical User Interface of FONASKEIN are given, along with its architecture. The evaluation results of FONASKEIN in a pilot experiment with eight participants and four songs in various musical scales showed its positive effect in practice of the singing voice in all cases.

1. INTRODUCTION

Singing practices in Modern Greece have a long history and display great diversity. Its roots go up to the interpretation of ancient Greek music which is considered as the theoretical fundament of Western music. The mathematical structure of Ancient Greek Music as referred to the works of Archytas, Philolaos, Didimos, Eratosthenis, Ptolemeos, and Aristoxenos still fascinates many researchers all over the world [1]. This written and oral tradition has been transferred to other types of music through the centuries such as the written theory of Byzantine music [2], the oral tradition of Greek folk music and even Rebetiko.

These unique characteristics of the diverse singing styles in Greece along with their mathematical relationships can be described in a generative way using the well-tempered tuning system; this causes confusion between the oral tradition and the music notation. Many of these different singing practices are carried out in Greek schools via traditional notation; the problem is that the teaching approach does not take into account the different tuning systems [3]. In this way the singing culture of children is still conflicted and depends on the cultural background of their family and the place of origin.

Although a number of visual feedback software applications for singing have been introduced in the recent years, non–equal tempered music scales are not a common feature of these software packages.

In this paper, we present the design, development, and evaluation of FONASKEIN, a novel modular interactive software application for the practice of singing voice in non-equal tempered scales.

2. STATE OF THE ART ANALYSIS

2.1 A quick Overview

One of the first attempts at designing software for the practice of the singing voice appeared in 1985 from G. Welch [4] who developed an innovative application for the BCC Microcomputer called SINGAD. SINGAD produced musical notes one-by-one, by recording the user's voice and analysing the recordings. The application compared the fundamental frequencies (F0) of the two signals and displayed the results on the screen.

At the beginning of the 1990s, Welch and his team improved this software and ran it on the Atari platform by improving it in three ways. First, instead of only comparing the fundamental frequencies of the two audio signals, SINGAD also compares the whole pitch contour which was more accurate. Secondly, SINGAD could play audio via MIDI synthesizers or sound from General MIDI (like piano or flute. Finally, the graphical user interface was made friendlier to musicians by including a viewer for the musical notes.

Another software application that was developed by Rossiter and his team in 1996 is called ALBERT [5]. Except the voice training, ALBERT included the monitoring of the laryngeal action. The system provided a greater variety of visual feedback by displaying the parameters F0, CQ (larynx closed quotient), spectral ratio, SPL (amplitude), shimmer and jitter. ALBERT was used in some studies in order to identify the quality of voice production during visual feedback implementation, and could measure the pattern of change during a training lesson.

Eight years later, in 2004, Callaghan and his team developed SING&SEE [6], one of the most popular

applications for the analysis of the singing voice with real time visual feedback (VFB). The main features of this research were the investigation of acoustic analysis technics, methods of displaying visual feedback in a meaningful way and the pedagogical approaches for implementing visual feedback technology into practice. Three parameters were distinguished as relevant for usage in the singing studio: pitch (F0 against time), vowel identity (R1, R2), and timbre (spectrogram). The major difference from previous studies was that not only quantitative but also qualitative data were of interest in this development.

In the same year, 2004, Welch and his team introdused a new project called VOXed. In this project Welch introduced the WinSINGAD [7]. The project also incorporated real-time VFB for singing education applications. While SING &SEE places emphasis on maximizing VFB technology itself, VOXed was aimed at maximizing the collaboration between different scientific fields. Psychologists, voice scientists, singing teachers, and singing students joined to form an interdisciplinary research team working for a better insight on the impact of VFB on the learning experience. Importantly, VOXed sought to work with participants as active agents rather than just passive recipients. The goal of the project was to investigate possible useful forms of VFB with the use of commercially available visual feedback software.

Another approach is the innovation of the MiruSinger software application developed by Nakako and his team [8] which introduced the possibility for the user to use an audio CD as a sample for comparison. MiruSinger analyzes the voice of the user, but also analyzes the voice from the song from the audio CD. Thus, it compares the audio signals from two human voices and not the human voice with a synthesized vocal sound. Nakako aimed to develop a software package for voice training with visual feedback with characteristics like tone accuracy, tempo, voice quality and expressive techniques (vibrato).

Lastly, the commercially available freeware Singing Coach[1] has been used in a number of studies in order to investigate children's voice profiles in a real educational environment; it has been tested in various countries and in Greek elementary schools where a computer-based vocal instruction methodology for music education has been proposed [3].

2.2 Critical approach

We appreciate that in the last thirty years there has been rapid evolution concerning the functionality and the incorporation of new parameters into the design of applications for the practice of the singing voice. For example, SINGAD uses only one parameter which is the detection of the fundamental frequency. ALBERT exploited the ever-increasing memory made common by the rapid development of personal computers in the 1990s. Furthermore, the advancement in combining different parameters for targeting different practices, such as singing and speech therapy has concretized the design of the software. SING&SEE mainly focused on aspects related to the same singers: fundamental frequency, identity vowel, and spectrogram. Then, the VOXed project introduced the WinSINGAD, which essentially combined the research parameters with those required by the musicians, namely the waveform, the fundamental frequency, various types of spectrograms in real time. Moreover, information captured by a camera was introduced for immediate feedback on the user's posture [9]. MiruSinger was considered innovative because it combined two real human voices with a reference to a voice recording from a commercial CD. Last, the Singing Coach software is more accessible and user friendly for children.

In general, optical feedback parameters have become more versatile and interdisciplinary over the years. Thus, these ameliorated software design principles opened access to a wider range of users. For example, SINGAD, in a first step has ben designed specifically for the development of children's voices, where ALBERT has been designed for wider applications and is not only for use in music education, SING&SEE and WinSINGAD have been specifically designed for singers of all ages and levels. Finally, all of them are being used by a variety of target groups.

3. THE FONASKEIN APPLICATION

FONASKEIN is a software application for real-time analysis of the singing voice with visual feedback. While the existing applications are limited to only two western scales (major and minor scales), FONASKEIN for the first time introduces the possibility to study and practice with non-equal tempered scales, such as the Byzantine or the ancient Greek scales. It also offers the user the option to enter a scale that is not included in the above or even to "build" their own scale. This can be achieved thanks to an "alteration mechanism" of each of the 12 notes to three semitones using cent resolution.

3.1 Design and Graphical User Interface

FONASKEIN was designed and implemented in Max/MSP. Thus, its GUI presented in Figure 1 was designed with the capabilities of Max/MSP and includes seven different windows.

The first window is the main bar at the top of the screen. It can hide or unhide other FONASKEIN's windows.

The second window is the audio control window. It is located on the left side of the screen. In this window, the user can control the audio input and output. Additionally, they can choose whether to record their voice or preview a prerecorded sample. Furthermore, the user can control the audio signal level both during both playback and recording.

The third window is the tuning window located on the right side of the screen. It includes an automatic tuner that indicates the deviation of the note that the user sings using a color scale.

[1] http://singingcoach.com/

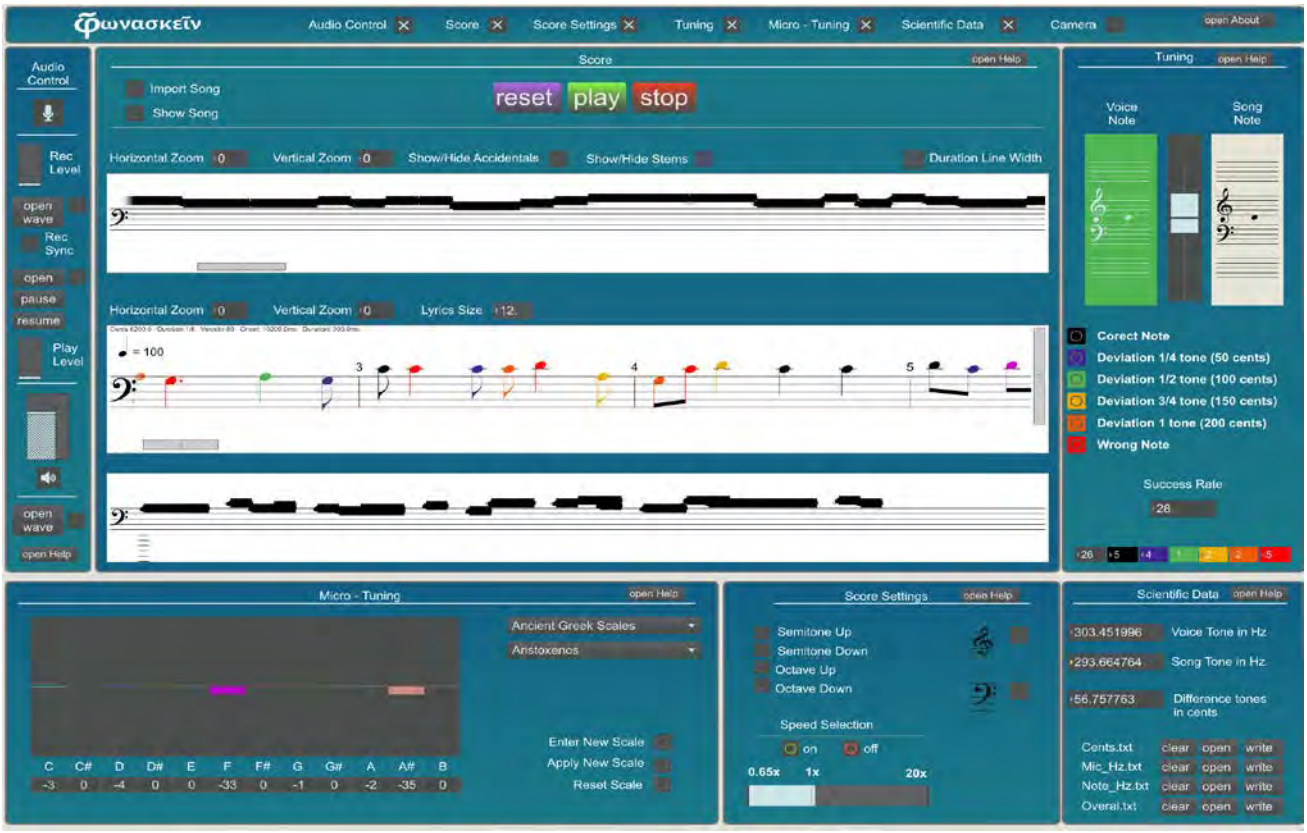


Figure 1. The Graphical User Interface of FONASKEIN.

The fourth window, the score window, is one of the most important windows. It is located in the middle of the screen and has two functions: a) it includes the main control buttons reset (play and stop). b) it presents three score windows where the user can read the musical piece with a piano roll view, regular view or see what they sang.

The next three windows are in the bottom of the screen and their main functions are the settings of FONASKEIN.

The window on the left side of the screen is a micro-tuning window. In this window, the user can select one of the default scales. There are three categories of scales, Western, Byzantine and Ancient Greek. Each of these has its own subcategories. The user can also import his/her own musical scales by writing the deviation of each note in cents under the multi-slider. FONASKEIN gives the user the possibility to play the song in these microtonal scales and listen to the correct musical intervals.

The next window is the score settings window. It is located in the middle of the screen and has three functions. The first one is the possibility to transpose the song a semitone lower, a semitone higher, an octave lower and an octave higher without affecting the micro tuning. The second possibility is to change the song's clef depending on the user's voice (bass clef for basses and tenors and treble clef for altos and sopranos). The last function is the speed selection, where the user can choose the speed of playback.

The last window is the data window which shows the current frequency that the user is singing, the current frequency of the correct note and the deviation in cents. The user has the possibility to view and save these data as *.txt files.

3.2 Architecture

The core of FONASKEIN comprises two parts. The first is related to the analysis and transformation of sound from the microphone signal and the second is dedicated to converting the MIDI file to a score as well as the import, playback, and control of microtonal scales.

For the first part, we used a Max Object called fiddle~. The operation of the algorithm of fiddle~ is based on the number of peaks of the audio signal where each one finds the tone of height and intensity. Specifically, the incoming signal is broken into segments of N samples with N a power of two typically between 256 and 2048. A new analysis is made every N=2 samples. For each analysis, the N samples are zero-padded to 2N samples and a rectangular window Discrete Fourier Transform (DFT) is taken using a rectangular window [10].

The next step is to calculate the frequency F0. Fundamental frequencies are guessed using a scheme somewhat suggestive of the maximum-likelihood estimator. The "likelihood function" is a non-negative function L(f), where f is the frequency. The presence of peaks at or near multiples of f increases L(f) in a way which depends on the peak's amplitude and frequency as shown:

$$L(f) = \sum_{i=0}^{k} a_i \, t_i n_i$$

where k is the number of peaks in the spectrum, a_i is a factor depending on the amplitude of the ith peak, t_i depends on how closely the ith peak is tuned to a multiple of *f*, and n_i depends on whether the peak is closest to a low or a high multiple of *f* [10].

The next step to build the FONASKEIN was the GUI score component. The Max/MSP does not support embedded objects with the creation pentagram, of notes and general notation. For this reason, we used not only an object designed by an external programmer, but a whole library comprising a large number of objects, the *bach library.*

The *bach library* is a cross-platform set of patches and externals for Max, aimed to bring the richness of computer-aided composition into the real-time world. In addition to that, it includes a large collection of tools for operating upon these new types and a number of advanced facilities and graphical interfaces for musical notation, with support for microtonal accidentals of arbitrary resolution, measured and non-measured notation, rhythmic trees and grace notes, polymetric notation, MusicXML and MIDI files [11].

As already stated, *bach* is a library of objects and patches for the software Max/MSP. At the forefront of the system are the bach.score and bach.roll objects. They both provide graphical interfaces for the representation of musical notation: bach.score expresses time in terms of traditional musical units and includes notions such as rests, measures, time signature, and tempo; bach.roll expresses time in terms of absolute temporal units (namely milliseconds), and as a consequence has no notion of traditional temporal concepts: this is useful for representing non-measured music, and also provides a simple way to deal with pitch material whose temporal information is unknown or irrelevant [12].

3.3 Non-equal tempered scales

One of the important novel features of FONASKEIN is its ability to import micro-tunings for singing in the Greek language. For the first time, the user is able to listen to a song that is written on a different scale from that of western music while he can exercise his voice on these interstices.

FONASKEIN, as mentioned above, includes a field with twelve sliders, one for each note. The sliders are able to move ± 300cents that can vary each note by three semitones. When the user presses the *Apply New Scale* button, a simple yet lengthy process allows the introduction of interstices of the two graphical objects of the *bach* library.

When the user changes the slider of a note by x cents, then the program will have to move all those notes in all octaves by the same distance. To do this it needs to follow a series of steps. The first step should be to choose the notes. After that, a second instruction enters the change of the note. This command is Cents = Cents + X. In this way, all selected notes have changed by the same pitch with cent accuracy. The time it takes FONASKEIN to do this is just 94 milliseconds, which is less than 1/10 of a second.

4. EVALUATION METHODOLOGY

The goal of the evaluation is to measure the change of the tonal errors in a singing voice by a number of participants after they practice with FONASKEIN in four songs with different music styles. The first song selected was “Ta paidia kato sto kampo” of Manos Hatzidakis (S1), a song written in the Western scale. The second song, “Thalassaki”, is a song in the Greek tradition scale Dorios (S2). The third song, “Apolitikion tou Staurou”, is a Byzantine hymn written in the First Mode (S3) and the last song, “Epitaph of Seikilos”, is an ancient Greek hymn written in 2nd century B.C. (S4). Eight postgraduate students of the University of Athens participated in the evaluation experiments. Among them, four were male and four female. Half of them were musicians.

The applied procedure follows the educational/training scenarios approach which is appropriate in testing computer-based tools in learning [13]. The educational scenario takes place through a series of educational activities. The structure and flow of each activity, the role of the learners in it and their interaction with the interactive software are described in the context of the scenario [14].

Two activities were included in our evaluation scenario, each with two tasks. In the first one each participant received four audio files made using FONASKEIN that correspond to the first seconds of the songs S1, S2, S3 and S4. The participants had to study themselves for a period of one week how to sing these songs, without any help. During the next task of this activity each participant sang the four songs he/she studied and the researcher digitally recorded their voices in a studio. Then the recordings were analyzed by FONASKEIN and the measured tonal errors constitute the comparison basis before the participants used FONASKEIN for training.

In the second activity the participants were asked to practice the four songs using FONASKEIN for the same period of one week. They fully exploited both its features of micro-tuning and the capability of visual feedback in real time. During the second task participants sang the four songs using FONASKEIN.

Finally, the participants completed a questionnaire with their demographic details, included both their cultural background and their relationship with the music and the four songs.

5. RESULTS

The analysis of the measurements in both activities was based on the following number of notes for each of the four songs: S1=61, S2=49, S3=66 and S4=37. We used MS-Excel 2010 for all the statistical analysis of the measurements.

Figure 2 presents for each one of the four songs S1-S4 the average of the positive and the negative errors in cents for all the participants and for all the notes for the two activities, i.e. before (b) and after (a) using FONASKEIN for the training of their singing voices, along with the standard error of the mean.

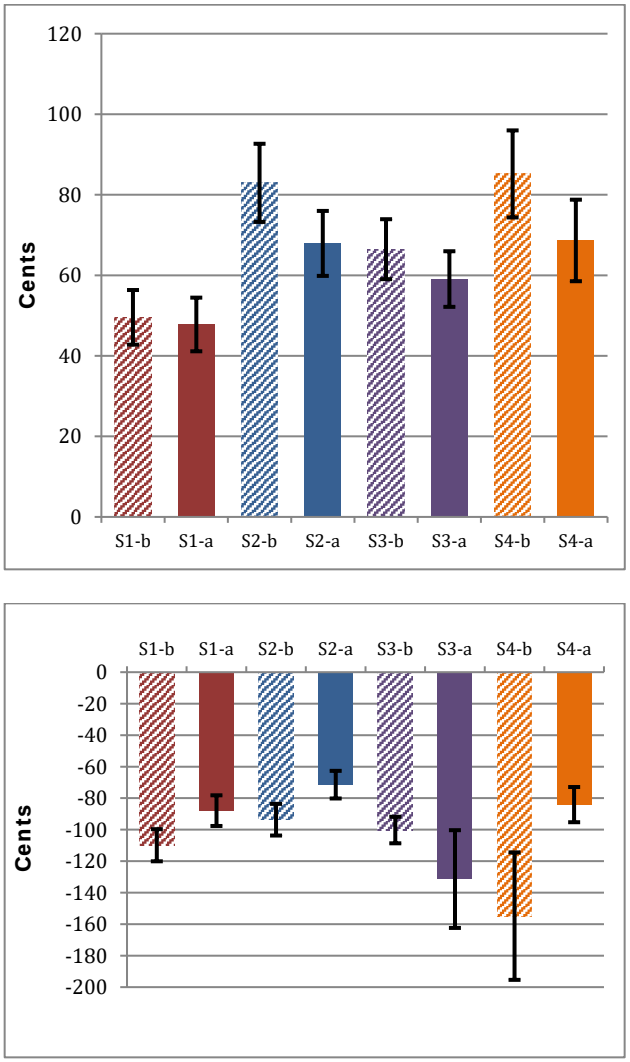


Figure 2. Average positive (above) and negative (bellow) errors in cents, for all the participants and for all the notes, before (b) and after (a) using FONASKEIN.

The number of negative errors was larger for all the songs. We observe a positive effect on using FONASKEIN as the number of errors was reduced in all the cases of the songs S1-S4. The largest improvement was for S4 (71 cents for the negative errors and 17 cents for the positive errors). The smallest improvement was for S1 (22 cents for the negative errors and 2 cents for the positive errors).

Figure 3 presents for each one of the four songs S1-S4 the average of the positive and the negative errors in cents for the participants who are musicians, for all the notes for the two activities, i.e. before (b) and after (a) using FONASKEIN for the training of their singing voices, along with the standard error of the mean.

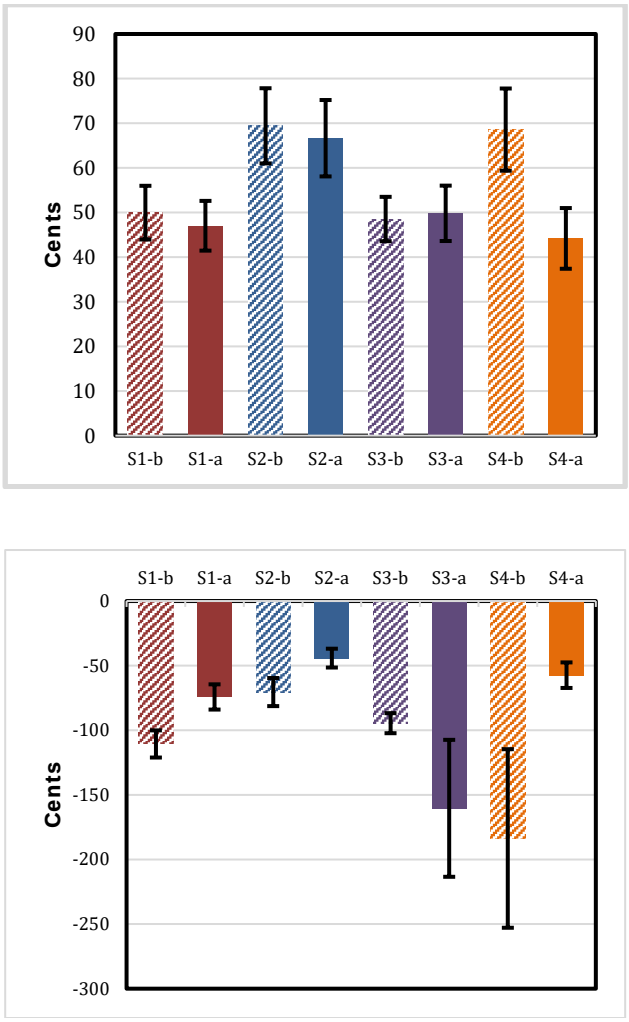
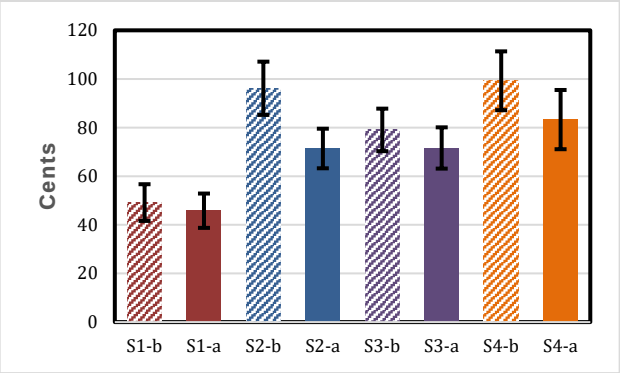


Figure 3. Average positive (above) and negative (bellow) errors in cents, for the participants who are musicians, for all the notes, before (b) and after (a) using FONASKEIN.

The number of negative errors was larger in almost all songs. We observed a positive effect on using FONASKEIN as the number of errors was reduced in all the cases of the songs S1-S4. The largest improvement was for S4 (126 cents for the negative errors and 24 cents for the positive errors). The smallest improvement was for S2 (3 cents for the negative errors and 27 cents for the positive errors).

Figure 4 presents for each one of the four songs S1-S4 the average of the positive and the negative errors in cents for the participants who are not musicians, for all the notes for the two activities, i.e. before (b) and after (a) using FONASKEIN for the training of their singing voices, along with the standard error of the mean. The number of negative errors was larger for all the songs. We observed a positive effect on using FONASKEIN as the number of errors was reduced in all the cases of the songs S1-S4, but much smaller compared to the relative for musicians. The largest improvement was for S2 (15 cents for the negative errors and 18 cents for the positive errors). The smallest improvement was for S1 (3 cents for the negative errors and 8 cents for the positive errors) and equally for S3 (7 cents for the negative errors and 4 cents for the positive errors).



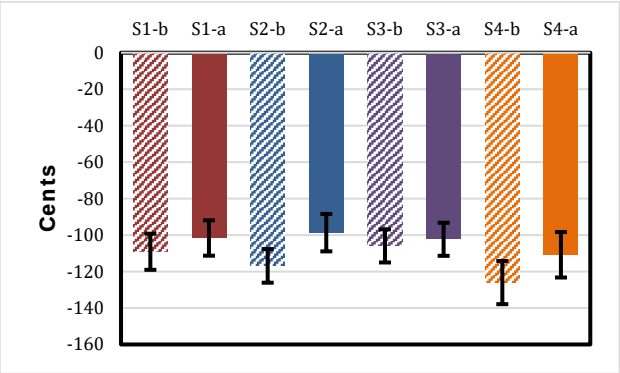


Figure 4. Average positive (above) and negative (bellow) errors in cents, for the participants who are not musicians, for all the notes, before (b) and after (a) using FONASKEIN.

6. CONCLUSIONS

We have presented the design and development of FONASKEIN, a novel modular interactive software application for the practice of singing in real-time and with visual feedback for both equal and non-equal tempered scales.

The evaluation results of FONASKEIN in a pilot experiment with eight participants and four songs in various musical scales showed its positive effect in practice of the singing voice in all cases.

In our future work we will study larger numbers of participants in various types of songs with non-equal tempered scales.

7. ACKNOWLEDGMENTS

We would like to aknowledge Panagiotis Velianitis, Professor Stelios Psaroudakes and George Chrysochoidis for their precious assistance.

8. REFERENCES

[1] M.L. West, and M. Litchfield, Ancient Greek music. Clarendon Press, 1992.

[2] M. Chrysanthos, Great Theory of Music. Michele Weis, 1832. Translated by Katy Romanou, The Axion Estin Foundation, New York, 2010.

[3] S. Stavropoulou, A. Georgaki, and F. Moschos, “The Effectiveness of visual feedback singing vocal technology in Greek Elementary School,” in Proc. Int. Computer Music Conference joint with Sound and Music Computing (ICMC|SMC|2014), Athens, 2014, pp.1768-1792.

[4] G. Welch, C. Rush, and D. Howard, “Realtime visual feedback in the development of vocal pitch accuracy in singing,” in Psychology of Music, vol 17, 1989, pp.146-157.

[5] D. Rossiter, and D. Howard, “ALBERT: real-time visual feedback computer tool for professional vocal development,” in Journal of Voice, vol 10, 1996, pp.321-336.

[6] J. Callaghan, W. Thorpe, and J. van Doorn, “The science of singing and seeing,” in Proc. Int. Conference on Interdisciplinary Musicology (CIM04), Graz, 2004.

[7] G. Welch, E. Himonides, D. Howard, and J. Brereton, “VOXed: Technology as a meaningful teaching aid in the singing studio,” in Proc. Int. Conference on Interdisciplinary Musicology (CIM04), Graz, 2004.

[8] T. Nakano, M. Goto, and H. Yuzuru, “MiruSinger: A Singing Skill Visualization Interface Using Real-Time Feedback and Music CD Recordings as Referential Data,” in Proc. Int. Conf. Ninth IEEE International Symposium on Multimedia, 2007, pp.75-76.

[9] D. Hoppe, M. Sadakata, and P. Desain, “Development of real-time visual feedback assistance in singing training: a review,” in Journal of Computer Assisted Learning, 2006, pp.308-316.

[10] S. Puckette, T. Apel, and D. Zicarelli, “Real-time audio analysis tools for Pd and MSP,” in Proc. Int. Conf. ICMC, Cologne, 1988.

[11] A. Agostini, and D. Ghisi, “bachproject,” http://www.bachproject.net/, Retrieved 24 April 2016.

[12] A. Agostini, and D. Ghisi, “Bach: an environment for computer-aided composition,” in Proc. Int. Computer Music Conf (ICMC2012), Ljubliana, 2012, pp.373-378.

[13] C. Kynigos, and E. Kalogeria “Boundary crossing through in-service online mathematics teacher education: the case of scenarios and half-baked microworlds,” in ZDM Int. J. on Mathematics Education, 2012, pp. 733-745.

[14] C. Kynigos, M. Daskolia, and Z. Smyrnaiou “Empowering teachers in challenging times for science and environmental education: Uses for scenarios and microworlds as boundary objects,” in Contemporary Issues in Education, 2013, pp. 41-65.

Visually Representing and Interpreting Multivariate Data for Audio Mixing

Josh Mycroft, Joshua D. Reiss, Tony Stockman
Centre for Digital Music, Queen Mary, University of London
j.b.mycroft@qmul.ac.uk

ABSTRACT

The majority of Digital Audio Workstation designs represent mix data using a channel strip metaphor. While this is a familiar design based on physical mixing desk layout, it can lead to a visually complex interface incorporating a large number of User Interface objects which can increase the need for navigation and disrupt the mixing workflow. Within other areas of data visualisation, multivariate data objects such as glyphs are used to simultaneously represent a number of parameters within one graphical object by assigning data to specific visual variables. This can reduce screen clutter, enhance visual search and support visual analysis and interpretation of data. This paper reports on two subjective evaluation studies that investigate the efficacy of different design strategies to visually encode mix information (volume, pan, reverb and delay) within a stage metaphor mixer using multivariate data objects and a channel strip design using faders and dials. The analysis of the data suggest that compared to channel strip designs, multivariate objects can lead to quicker visual search without any subsequent reduction in search accuracy.

1. INTRODUCTION

The majority of Digital Audio Workstation (DAW) designs represent mix data using a channel strip metaphor where individual controls are mapped on a one-to-one basis to mixing parameters. So, for example, equalisation, pan position, volume and effects (such as reverb) are all represented by different virtual controls. This can result in an increasingly complex interface [1, p.1] leading to a fragmented and disjointed approach to mixing [2]. Furthermore, the use of dials to represent the mix information (a major design element of channel strip designs) can be hard to interpret due to the fact that the human eye has difficulty comparing angles, specifically underestimating acute angles and overestimating obtuse angles [3 p. 49].

Within other areas of data visualisation (such as medial visualisations, geo-spatial and cartographic displays) many-to-one mappings are used to simultaneously represent a number of parameters within one graphical object, by assigning data to specific visual variables such as position size, shape, hue, saturation, texture, opacity and dynamics [4,5,6]. This can reduce screen clutter, help support the interpretation of data and enhance visual analysis by allowing both inter and intra-record relationships to be more easily detected [7]. Indeed, research by Dewey et al, [8] has shown that the use of icon based mixers can not only reduce cognitive load on the user but also increase immersion. However, due to the limits of human visual perception, there are constraints on the design of multivariate data objects [5]. For example, while colours can be interpreted easily when displayed at reduced sizes they are liable to certain caveats such as the range of colours that can be effectively differentiated and the potential issue of 'colour blindness' among users [9]. Furthermore, some studies suggest that visually representing several streams of information at the same time can increase cognitive processing load [10,11].

In DAW design alternatives to the channel strip metaphor exist which make use of many-to-one mappings. For example, in [12] a virtual microphone position is used to represent the relative fader settings for multiple microphones around a sports stadium. Another alternative is the *stage metaphor,* a design which visualises channels as sound sources on a virtual stage where one can control pan position (relative left right position in the stereo field) and volume within a single User Interface (UI) object using its *x* and *y* positions [13,14,15]. Previous work by the authors has found that the consequent reduction in UI objects can minimise the need for navigation, allow significantly quicker visual search of mix parameters and improve concurrent critical listening tasks compared to an equivalent channel strip design [16]. However, a typical channel strip mixer will represent equalisation and audio effects as well as pan and volume position [17]. Being able to represent these within a stage metaphor design is therefore necessary in order to convey important attributes of the mix.

This paper therefore evaluates the efficacy of different designs to visually represent further mix parameters within a stage metaphor mixer by assigning mix parameters to multivariate data objects and comparing the visual search

times and accuracy to a channel strip mixer. By so doing the authors hope to convey mix information in a way which is perceptually and cognitively efficient and which optimally supports the interpretation of visual mix data.

2. STUDY ONE: REPRESENTING AN ADDITIONAL MIX PARAMETER

2.1 Participants

Participants were comprised of staff and students on a two-year music technology course at City and Islington College, London. All participants had at least one year's experience mixing on DAWs (with a minimum of five hours a week exposure to DAWs and mixing). Sixteen participants were selected (10 male, 6 female aged 17-19). The details of the study were approved by the ethics department of QMUL.

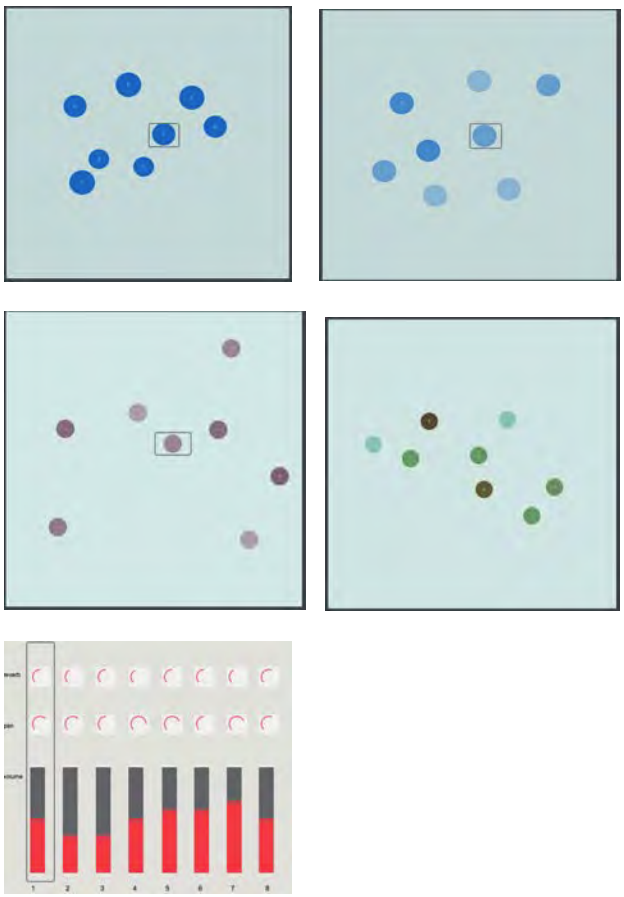


Figure 1. Screens for Study One: (a) top left; size, (b) top right; transparency, (c), middle left, saturation, (d) middle right, colours (e) bottom, the channel strip mixer.

2.2 Study Design

Five eight-channel mixers; a channel strip design and four stage metaphor mixers (figures 1, a-e) were designed using Max/MSP showing each channel's volume, pan and reverb amount (reverb is a commonly used audio effect which is often used to simulate real acoustic space, giving sounds a sense of ambience in the mix). As the visual representation and interpretation of the mix data was the object of the investigation, no audio was used; each mixer design was a visual representation only. The term reverb was used solely to contextualise the visual tasks and place the additional parameter within an audio mixing frame of reference.

For the channel strip design, faders were used for volume, while dials were used for the pan position and the reverb. For the stage metaphor, *x* and *y* positions were used for the pan and volume, while five designs were used to represent the reverb: size, transparency, saturation (single colour) and hue (multiple colours). Rate of flashing (dynamics) was not used due to concerns that this might trigger seizures among people with photosensitive epilepsy [18]. Shading was discounted due to the difficulty of interpretation at the high zoom levels required to analyse the overview [19] and shape was not included since it is chiefly a categorical data set [20].

The objective of the study was to ascertain how subtle a difference could be visually perceived between channels with different reverb amounts and how fine a range of values could be represented using each design. In order to do this, the reverb's range (1-100) was divided into increments of five, ten and twenty values and assigned to each design. Increments of less than five were not included due to perceptual issues; colour schemes divided into multiple steps become increasingly hard to differentiate, with the values represented becoming difficult to distinguish [21]. Furthermore, some displays will not accurately display small colour differences due to varying visual display characteristics (ibid).

To represent increments of five reverb values using colour and saturation, twenty gradients were used (fig 2); gradient 1 showing reverb values of 1-5, gradient 2 showing 5-10 etc. For increments of ten, alternate gradients were used with each one representing a range of ten values (0-10, 10-20 etc.). For increments of twenty, only five gradients were used, each representing a range of twenty values (0-20, 20 40 etc.). In all cases darker colours were used to represent less reverb. For size, the difference between the minimum and maximum circle diameter was divided into five, ten and twenty sizes. To represent increments of five reverb values, twenty circle sizes were used (the smallest circle showing values of 1-5, the second smallest showing value 5-10 etc.), to represent increments of ten, alternate sizes were used (each depicting a range of 10 values) and to represent increments of twenty, five circle sizes were used (each representing a range of 20 values). In all cases smaller circle sizes represented less reverb. Finally, the same method was used for transparency; the most and least transparent settings were divided into 5, 10 and 20 differences and assigned to reverb amounts with the most transparent settings representing the most reverb.

For each of the five mixer designs (channel strip, size, colour, saturation and transparency) a target was included in the eight channels and placed within a border (fig 1). For each design three screens were created; one with reverb differences between the target and other channels set at +/- 5 (increments of 5), one with differences set at +/- 10 (increments of 10) and one with differences at set +/- 20 (increments of 20). This created a total of fifteen screens for the study.

2.3 Procedure

Each participant was presented with each mixer design at the three increment differences between target and other channels (which were randomized for each participant). This meant that, for example, on the screens showing increments of 5, if the target reverb value were set to 50, the other channels would all be 45 or 55 with the exception of one other channel that was also set to the target's value. For each screen participants were asked to identify which of the other channels on the mixer had the same reverb value as the target channel by clicking on the corresponding channel. The screen order was randomized for each participant and they were presented one after the other.

The mapping of the designs to reverb amount (e.g. smaller circle size to less reverb) was explained to each participant and they were given time to familiarise themselves with the different interface designs using practice screens. Participants were asked if they suffered from any known form of colour blindness prior to the test (no respondents reported this). Immediately after the study each participant was asked about their experience of using the different designs.

Figure 2. Colour gradients used in the Studies: Top; single colour saturation (less saturated colours were mapped to greater reverb amounts). Bottom; colours (darker colours were mapped to less reverb).

2.4 Analysis and Results

The amount of errors (incorrectly identified channels) was calculated for each participant in each of the fifteen interfaces. From this the total number of errors made on each screen by all participants could be calculated (table 1). The results show that within all designs the error rates increased as the visual differences between the target and other channels' reverb values became smaller. However, the most errors for all differences were found in the dials and transparency designs. Size, colour and saturation resulted in fewer errors even at smaller differences.

In order to test the significance of the error rates found between the different mixers, the data was analysed using a *z*-test for proportions dependent groups at 95% Confidence intervals (CI). The results of the analysis show that the difference between the dials and transparency compared to the other designs was significant for increments of 5 and 10 per cent differences. However, the analysis showed no significant difference in accuracy between size, colours and saturation (though size had the least errors).

Increments between target and other channels' reverb amounts	*5*	*10*	*20*
dial	68	50	18.7
colour	25	18.7	12
saturation	25	18.7	6.2
size	18.7	6.2	6.2
transparency	68	65	31.2

Table 1. Error rates (%) for each design at different value differences between target and other channels. Correctly identifying similarity between the channels was worst for the dial and transparency designs at all increment differences. Size proved the least error prone, with saturation and colour being generally evenly matched.

The participants were also asked to comment on using the different designs. Several of the participants said they found the transparency design very difficult, as it was hard to tell the difference between the reverb values, even at differences of 20%. A source of confusion for the colour design was the mapping of the reverb values; a number of participants expressed confusion over which way it was mapped, e.g. did lighter colours represent more or less reverb. This issue did not occur with size, where all participants were happy with the "bigger is more" metaphor. This was also less of a problem with the saturation of the single colour where less saturated was more readily understood as representing more reverb.

3. STUDY TWO: ADDING A FURTHER MIX PARAMATER

3.1 Participants

Participants were comprised of staff and students on a two-year music technology course at City and Islington College, London. All participants had at least one year's experience mixing on DAWs (with a minimum of five hours a week exposure to DAWs and mixing). For Study

Two, twelve participants were selected (7 male, 5 female aged 17-35). Separate participants were used for Studies One and Two to avoid the risk of possible learning effects. The details of the study were approved by the ethics department of QMUL.

3.2 Study Design

This study was designed to evaluate the efficacy of adding two mix parameters (reverb and delay) in addition to panning and volume. Again this was done using both channel strip and stage metaphor designs. As with Study One, no audio was used, as the aim of the study was to evaluate the efficacy of visual representation and interpretation. As with Study One, the terms reverb and delay were used to place the visual tasks within an audio mixing context, rather than specifically assessing these audio effects.

The choice of visual designs for the study was based on the results from Study One. As outlined in section 2.4, size had performed best, while colour and saturation had been evenly successful. Transparency however had shown a significantly higher error rate (table 1); a result which corresponds with research suggesting that colour and size are the dominant visual channels and are most efficiently interpreted [6]. Transparency therefore was discounted for Study Two. Between colour and saturation, the latter was taken forward due to it being a colour-blind safe design and due to the fact that multiple colours had resulted in some confusion from users over mapping.

Again, a channel strip design using faders and dials was included so that a direct comparison could be made between design outcomes of multivariate objects and current design paradigms. For the stage metaphor design, *x*-axis and *y*-axis were linked to pan and volume while reverb was linked to size and delay linked to saturation. As with Study One, the reverb and delay parameters were given values of 100 steps, and the mixers represented these in increments of 20, 10 and 5 divisions.

3.3 Procedure

Participants were presented with both designs of an eight-channel mixer (figure 3) and were asked to identify a particular channel in relation to the target channel (surrounded by a border). For example, they were asked which channel was panned left of the target, of a higher volume than target, with the same amount of reverb and less delay than target? The task was chosen as it required the simultaneous analysis of all four visual channels (*x* and *y* position, size and saturation).

There were 18 screens in total. Nine stage metaphor screens and nine channel strip screens. Both designs included three screens with 5% differences between the target and other channels' delay and reverb settings, three with 10% difference, and three with 20% differences. So, for example, if target had a setting of 50 on reverb and 75 on delay, the 5% difference would mean the other tracks were set to reverb being either 45 or 55 and delay between 70 or 80, with the exception of one other channel which was assigned the same reverb and delay settings as the target.

The order in which the mixers were presented was randomised for each participant. The reverb and delay values of the other seven channels were randomised for each participant (within variations of 5, 10 or 20 increments). The channel(s) chosen and the time taken to choose them were recorded for each participant, though this was not visible to them. Participants were given time to familiarise themselves with the mixer designs using practice screens before beginning the evaluation.

3.4 Results and Analysis

The amount of errors (incorrectly identified channels) were calculated for each participant in all 18 screens. From this the total number of errors made on each screen by all participants could be calculated (table 2). The results show that the percentage of errors in selecting the correct channel was higher when analysing the mix using faders and dials than the multivariate data objects.

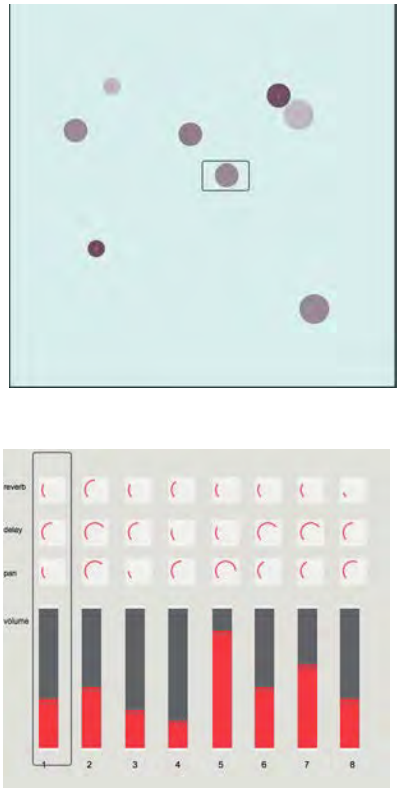
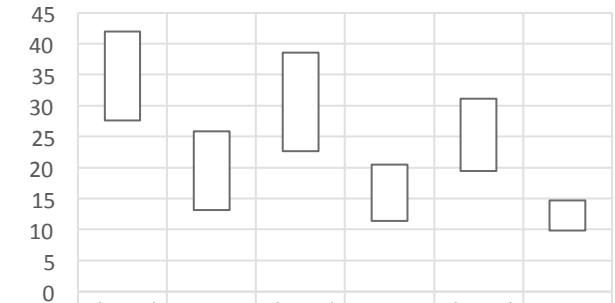


Figure 3. Top (a), the stage metaphor mixer; *x* and *y* positions show pan and volume, saturation of red colour shows delay amount and size shows reverb amount. Bottom (b) channel strip mixer; faders show volume, dials show pan, reverb and delay.

These results were analysed using a *z*-test for proportions to see if the error rate between designs was significant (at 95% CI). The analysis revealed that while the error rates for the multivariate design were much lower than the channel strip at 5% differences, there was no statistical difference between the two, which may be due to the increased visual load required to analyse colour, size and position attributes simultaneously [10,11]. At 10% increments, however, there was a significant difference in error rates in favour of the stage metaphor design. As with study one, no significant difference was found at 20 % differences, possibly due to the fact the perceptual difficulties found in estimating angles in dials ceased to be an issue when the difference was increased to this level.

The time to identify the correct channel was also analysed for each participant in both mixer designs at the different increment levels. From this the mean time and standard deviation were calculated. This was used to generate Confidence Intervals at 95%. The analysis revealed significant time differences in identifying the correct channels between the channel strip and stage metaphor designs with the former taking significantly longer at all increment levels (figure 4). The analysis suggests that the stage metaphor multivariate mixer allows users to find visual information significantly quicker without any subsequent increase in error rate.



	channel strip 5	stage 5	channel strip 10	stage 10	channel strip 20	stage 20
High	27.6	13.11	22.6	11.4	19.4	9.8
Low	42	25.8	38.5	20.5	31.1	14.7

Figure 4. The visual search time (seconds) was significantly quicker in the stage metaphor design. In both designs search times decreased as differences between channels became greater.

Increments	*5*	*10*	*20*
Channel strip	36.1%	33.3 %	11.1%
Stage metaphor	11.1%	5.5%	5.5%

Table 2. Error rates for both design at different value differences. The stage metaphor design was significantly more accurate at increments of 10 percent.

4. CONCLUSION AND FUTURE WORK

The results of these studies suggest that mapping mix attributes within a single multivariate object can result in significant improvements in visual search time and accuracy compared to a channel strip design. The multivariate designs allowed users to find four separate mix parameters (pan, volume, reverb and delay) more rapidly within one UI object than the four UI objects required in a channel strip design. Given the increase in mix capacity and the reduction in screen size found in tablet computers this may prove useful in reducing screen clutter and helping users better analyse and interpret the visual information presented.

However, the results of these studies suggest that the design of the visual channels used to encode additional mix parameters must be perceptually suitable, and cannot be assigned in an arbitrary manner. For example, multiple colours caused confusion over mapping, while transparency became difficult to interpret at reduced values. However, while not all visual channels used in the studies were equally effective, there may still be uses for them. For example, transparency may be useful for showing coarser values, such as muted and unmuted channels or indicating occlusion in mixes where channels visually overlap [21]. Multiple colours, while prone to mapping confusion, may be suitable to more ordinal tasks such as identifying which channels are grouped together (e.g. vocals, drums, percussion instruments etc.) [23]. Furthermore, the relative novelty of the colour mappings in this study may be a factor in confusion, and prolonged use may lead to a greater acceptance as mapping schemes become better understood [6, p.2].

The lack of significant improvement in error rates between the multivariate designs and channel strip designs at 5% increments may have been due to the increased visual load required to analyse colour, size and position attributes simultaneously [10,11]. Previous work by the authors has shown that the use of Dynamic Query (DQ) filters (UI objects such as sliders that facilitate real time visual display of query formulation and results) resulted in a higher amount of correctly completed visual and aural tasks compared to versions of the same interface without them [24]. DQ filters may be applicable to displaying multivariate data; allowing the user to visually explore and filter the information while continuously viewing the changing results.

Lastly, the authors acknowledge that this paper is preliminary in the sense that it focuses exclusively on visual aspects. Future studies should incorporate audio tasks alongside existing and multivariate designs to assess the extent to which they ameliorate potential difficulties in simultaneously analysing multiple data and help keep the users' attention optimally focused on interpreting both visual and auditory mix data.

5. REFERENCES

[1] K. Golkhe, M. Hlatky, S. Heise, D. Black, J. Loviscach. "Track Displays in DAW Software: Beyond Waveform Views". In: *Proc. Audio Engineering Society,* London, 2010.

[2] J. Mycroft, T. Stockman, J.D. Reiss. 'The Influence of Graphical User Interface Design on Critical Listening Skills'. In: *Sound and Music Computing (SMC),* Stockholm, 2013.

[3] N. Robbins. *Creating More Effective Graphs*. Wiley, 2005.

[4] J. Bertin. *Semiology of Graphics.* University of Wisconsin Press, Madison WI 1981.

[5] W. Cleveland. *Visualizing Data*. Hobart Press, Summit, New Jersey, 1993.

[6] R. Borgo, J. Kehrer, D. H. Chung, E. Maguire, R. S. Laramee, H. Hauser, M. Ward, and M.Chen. "Glyph-based visualization: Foundations, design guidelines, techniques and applications". In: *Eurographics 2013–State of the Art Reports,* pages 39–63. The Eurographics Association, 2012.

[7] M. Ward. Multivariate data glyphs: "Principles and Practice". In: *Handbook of Data Visualisation*, 2008.

[8] C. Dewey and J. Wakefield. 'Novel designs for the audio mixing interface based on data visualaisation forst pronciples'. In: *Proceedings of the 140th AES Convention*, Paris, June 4-7, 2016.

[9] M. Stone . "Choosing Colors for Data Visualization" http://www.perceptualedge.com/articles/b-eye/choosing_colors.pdf. Accessed March 2016.

[10] R. Gudur, A. Blackler, V. Popovic and D.P Mahar, D. P. "Redundancy in interface design and its impact on intuitive use of a product in older users." In: *International Association of Societies of Design Research Conference, Seoul* 2009.

[11] S. Gelineck and D. Overholt. 'Haptic and Visual Feedback in 3D Audio Mixing Interfaces'. In: *Proceedings of the 9th Audio Mostly Conference,* Thessaloniki, Greece, 2015.

[12] H. Bourne, J. D. Reiss, "Evaluating iBall—An Intuitive Interface and Assistive Audio Mixing Algorithm for Live Football Events," 135th AES Convention, New York, Oct., 2013

[13] C. Dewey and J. Wakefield. 'A guide to the design and evaluation of new user interfaces for the audio industry'. In: *136th Audio Engineering Society Convention*, 26th-29th, Berlin, Germany, April 2014

[14] S. Gelineck and D. Korsgaard. "Stage metaphor mixing on a multi-touch tablet device". In *Audio Engineering Society Convention*, 2014.

[15] P. Gibson. *The Art Of Mixing: A Visual Guide To Recording, Engineering, And Production*. ArtistPro Press, 1997.

[16] J. Mycroft, J.D. Reiss, T. Stockman, T. "The effect of differing user interface presentation styles on audio mixing." In: *International Conference on the Multimodal Experience of Music (ICMEM),* Sheffield, 23-25 March 2015.

[17] M. Cartwright, B. Pardo and J.D.Reiss. "Mixploration: Rethinking the audio mixer interface". In: *19th Int. Conf. on Intelligent User Interfaces,* pages 365–370, Jan. 2014.

[18] About Epilepsy. Available:http://epilepsy.org/about-epilepsy/.org.

[19] M. Hlatky, K. Gohlke, D. Black and J. Loviscach. "Enhanced Control of On-Screen Faders with a Computer Mouse". In: *Proc. Audio Engineering Society.* Munich, Germany 2009.

[20] D. Chung, R. Laramee, J. Kehrer, H. Hauser and M. Chen. "Glyph-based Multifield Visualisation". In: *Scientific Visualisation*, Springer 2014.

[21] M. Harrower and B. Sheesley. "Designing Better Map Interfaces: A Framework for Panning and Zooming". In: *GIS*, 9, 77–89, 2005.

[22] H. Leitte and C.Heine. *Visual Data Analysis*. http://www.iwr.uni_heidberg.de/groups/CoVis. Pdf.

[23] D. M. Ronan, B. De Man, H. Gunes and J. D. Reiss. 'The impact of subgrouping practices on the perception of multitrack mixes,'. In: *139th AES Convention,* NY, 2015

[24] J. Mycroft, T. Stockman, J.D. Reiss 'Visual information Search in Digital Audio Workstations'. In*: Proceedings of the 140th AES Convention*, Paris, June 4-7, 2016.

RHYTHM TRANSCRIPTION OF POLYPHONIC MIDI PERFORMANCES BASED ON A MERGED-OUTPUT HMM FOR MULTIPLE VOICES

Eita Nakamura
Kyoto University
enakamura@sap.ist.i.kyoto-u.ac.jp

Kazuyoshi Yoshii
Kyoto University
yoshii@kuis.kyoto-u.ac.jp

Shigeki Sagayama
Meiji University
sagayama@meiji.ac.jp

ABSTRACT

This paper presents a statistical method of rhythm transcription that estimates the quantised durations (note values) of the musical notes in a polyphonic MIDI performance (*e.g.* piano) signal. Hidden Markov models (HMMs) have been used in rhythm transcription to combine a model for music scores and a model describing the temporal fluctuations in music performances. However, when applied to polyphonic music, conventional HMMs have a problem that they are based on representation of polyphonic scores as linear sequences of chords and thus cannot properly describe the structure of multiple voices. We propose a statistical model in which each voice is described with an HMM and polyphonic performances are described as merged outputs from multiple HMMs, based on the framework of merged-output HMM. We develop a rhythm-transcription algorithm based on this model using an efficient Viterbi algorithm. Evaluation results showed that the proposed model outperformed previously studied HMMs for rhythm transcription of polyrhythmic performances.

1. INTRODUCTION

Music transcription is a fundamental problem in music information processing, requiring the extraction of pitch and rhythm information from music audio signals. There have been many studies on converting a music audio signal into a piano-roll representation based on acoustic modelling of musical sound [1, 2]. To obtain a music score, we must recognise quantised note lengths (or note values) of the musical notes in piano rolls. For this purpose, many studies have been devoted to solving the problem of converting MIDI performances to music scores, which is called rhythm transcription or quantisation [3–12]. In accordance with the general trend, statistical modelling has been gathering attention recently in this field.

Hidden Markov models (HMMs) [13] are the most popular models used in recent studies on rhythm transcription [5–10]. Indeed a monophonic score, when represented as a series of musical notes, can naturally be described with a Markov model. In addition, temporal fluctuations in performances can be described by a continuous-space HMM with a latent variable corresponding to time-varying tempos [10, 14, 15].

When HMMs are used for modelling polyphonic music, we immediately face the problem of score representation. A polyphonic score has multilayer structure, where concurrently sounding notes are grouped into several streams or, in music terminology, *voices* [1]. A conventional way is to represent a polyphonic score as a linear sequence of chords [7]. However, this representation may not retain sequential regularities within voices, such as those in polyrhythmic scores. Furthermore, properties of music performance, like the phenomenon of loose synchrony between voices [17, 18], cannot be captured without explicitly modelling the multiple-voice structure.

The purpose of this paper is to construct a statistical model for rhythm transcription that can describe the multiple-voice structure of polyphonic music scores and performances. We construct a model that describes polyphonic performances as merged outputs from multiple component HMMs, each of which describes the generative process of music scores and performances of one voice. Our model is based on the merged-output HMM [19, 20], which has been developed to describe, in an event-driven manner, symbolic data of polyphonic music. We derive an efficient inference algorithm that can simultaneously separate performed notes into voices and estimate their note values. The proposed model is compared with previously studied HMM-based models by evaluating the accuracy of rhythm transcription for piano performances. A complete model description and extended evaluation results will be presented in our forthcoming paper [23].

The main contribution of this study is the construction of a rhythm-transcription algorithm that can explicitly handle multiple voices with guaranteed optimality. A statistical model with multiple-voice structure based on two-dimensional probabilistic context-free grammar (PCFG) models has been studied [11, 12], but the algorithms developed in those studies had to use provided voice information or a pruning technique that would sacrifice optimality.

2. RELATED WORK

In this section, we review previous HMM-based models for rhythm transcription and discuss the problem of polyphonic extensions.

[1] In this paper, a 'voice' means a unit stream of musical notes that can contain chords.

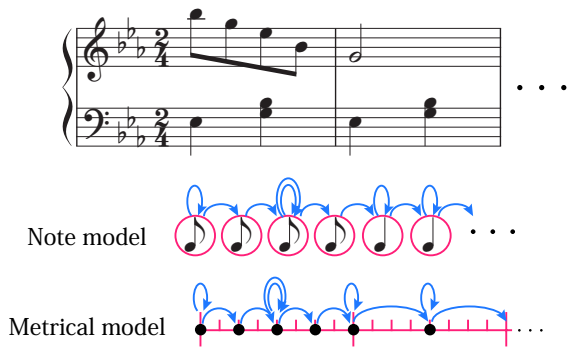


Figure 1. Two different representations of a music score in previously proposed HMMs.

2.1 HMM-Based Models for Monophonic Music

HMMs for rhythm transcription usually consist of two component models; a score model describing the probability of a score and a performance model describing the probability of a performance given a score. HMMs in previous studies [5–10] can be classified into two groups according to the way the score model describes the sequence of notes. In one class of HMMs for rhythm transcription, which we call *note HMMs*, a score is represented as a sequence of note values and described with a Markov model (Fig. 1) [5,6]. To describe the temporal fluctuations in performances, one introduces a latent variable corresponding to a (local) tempo that is also described with a Markov model. An observed duration is described as a product of the note value and the tempo that is exposed to noise of onset times.

In another class of HMMs, which we call *metrical HMMs*, a different description is used for the score model [8–10]. Instead of a Markov model of note values, a Markov process on a grid space representing beat positions of a unit interval, such as a bar, is considered (Fig. 1). The note values are given as differences between successive beat positions. The same performance model as in note HMMs can be used. Incorporation of the metre structure is an advantage of metrical HMMs.

2.2 Polyphonic Extensions

There are two directions of polyphonic extensions: using a simplified representation of polyphonic scores or using an extended model describing multiple voices. The first direction is based on a fact that any polyphonic score can be represented as a sequence of chords or, more precisely, 'note clusters' consisting of one or more notes as far as only onsets are concerned. For note HMMs, chordal notes can be represented as self-transitions in the score model (Fig. 1) and their inter-onset intervals (IOIs) can be described with a probability distribution with a peak at zero [7]. Similar extensions are possible for metrical HMMs.

For the second direction, a PCFG model has been extended to describe the multiple-voice structure of scores [11]. In addition to the divisions of a time interval, duplications of intervals into two voices are considered. Unfortunately, a tractable inference algorithm could not be obtained for this model, and the correct voice information had to be provided for evaluations. Takamune *et al.* state that this problem is solved using the generalised LR parser [12]. Although detailed explanations are lacking, their method uses pruning and its optimality is not guaranteed.

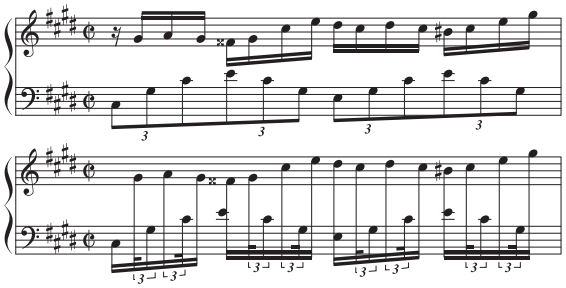


Figure 2. A polyrhythmic passage (Chopin's Fantaisie Impromptu) represented as a sequence of chords.

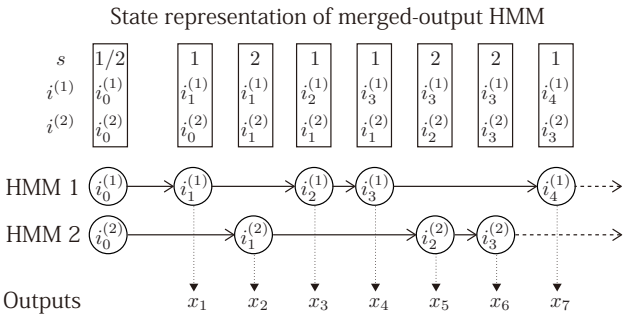


Figure 3. A schematic illustration of the merged-output HMM. The symbols $i_0^{(1)}$ and $i_0^{(2)}$ represent auxiliary states to define the initial transitions.

Although the above two descriptions of polyphonic scores are both logically possible, there are instances in which models based on the simplified representation cannot describe the nature of polyphonic music well. First, complex polyphonic scores such as polyrhythmic scores are forced to have unrealistically small probabilities. This is because such scores consist of rare rhythms in the simplified representation even if the component voices have common rhythms (Fig. 2). Second, the phenomenon of loose synchrony between voices (*e.g.* two hands in piano performances [17]), called *voice asynchrony*, cannot be described. Indeed, the importance of incorporating the multiple-voice structure in describing polyphonic music is well-established in studies on score-performance matching [17, 18]. The situation calls for a similar treatment of multiple voices for polyphonic rhythm transcription.

2.3 Merged-Output HMM

Recently merged-output HMM has been proposed as an HMM-based model for describing symbolic signals of polyphonic music with multiple voices. In the model, each voice is described with an HMM and the total signal is represented as merged outputs from these HMMs (Fig. 3). The merged-output HMM can be seen as a variant of factorial HMM [21]. To appropriately describe the nature of symbolic signals and capture sequential regularities within each voice, only one of the component HMMs is involved with each output in a merged-output HMM, whereas all component HMMs contribute to every output in a standard

factorial HMM. Basic inference algorithms for merged-output HMMs have been provided in our previous studies [19, 20].

3. PROPOSED MODEL

We present an HMM-based model for rhythm transcription that describes polyphonic performances with multiple-voice structure. Given a polyphonic MIDI performance signal, the model can simultaneously separate performed notes into voices and estimate their note values. To construct a model based on a previously studied HMM [7] and apply the framework of merged-output HMM [19, 20], we address the following issues: (1) pitches should be explicitly modelled to appropriately describe voices; (2) tempos of multiple voices should be bound to assure loose synchrony between voices. After explaining the note HMM in detail in Sec. 3.1, a model satisfying these requirements is presented in Sec. 3.2, and a sketch of inference algorithm is given in Sec. 3.3.

A music score is specified by multiple sequences, corresponding to voices, of pitches and note values. Since polyrhythm and voice asynchrony typically involve two voices, we formulate the model with two voices indexed by a variable $s = 1, 2$. A MIDI performance signal is specified by a sequence of pitches and onset times.

3.1 Model for Each Voice

For each voice we first construct a model based on the one presented in a previous study [7]. Let N_s be the number of score notes in voice s and let $r_n^{(s)}$ denote the note value of the n-th note. The note values $\boldsymbol{r}^{(s)} = (r_n^{(s)})_{n=1}^{N_s}$ are generated by a Markov chain with the probability given as

$$r_1^{(s)} \sim \mathsf{Cat}(\boldsymbol{\pi}_{\mathrm{ini}}^{(s)}), \tag{1}$$

$$r_n^{(s)} | r_{n-1}^{(s)} \sim \mathsf{Cat}(\boldsymbol{\pi}_{r_{n-1}^{(s)}}^{(s)}) \quad (n = 2, \ldots, N_s), \tag{2}$$

where Cat denotes the categorical distribution, $\boldsymbol{\pi}_{\mathrm{ini}}^{(s)} = (\pi_{\mathrm{ini},r}^{(s)})_r$ is the initial probability, and $\boldsymbol{\pi}_{r_{n-1}^{(s)}}^{(s)} = (\pi_{r_{n-1}^{(s)},r}^{(s)})_r$ is the (stationary) transition probability. Chordal notes are represented as self-transitions of note values (Fig. 1). The probability values are to be learned from music data.

To describe the temporal fluctuations, we introduce a tempo variable, denoted by $v_n^{(s)}$, that describes the local tempo for the n-th note. To represent the variation of tempos, we put a Gaussian Markov process on the logarithm of the tempo variables as

$$\ln v_n^{(s)} | \ln v_{n-1}^{(s)} \sim \mathsf{N}(\ln v_{n-1}^{(s)}, \sigma_v^2), \tag{3}$$

where N denotes the normal distribution. If the $(n-1)$-th and n-th notes belong to a chord, their IOI approximately obeys an exponential distribution [15] and the probability of the onset time of the n-th note, denoted by $t_n^{(s)}$, is then given as

$$t_n^{(s)} | t_{n-1}^{(s)} \sim \mathsf{Exp}(\lambda), \tag{4}$$

where Exp denotes the exponential distribution and λ is the scale parameter. Otherwise, $t_n^{(s)} - t_{n-1}^{(s)}$ has a duration corresponding to note value $r_{n-1}^{(s)}$ and the probability is described with a normal distribution as

$$t_n^{(s)} | t_{n-1}^{(s)}, v_{n-1}^{(s)}, r_{n-1}^{(s)} \sim \mathsf{N}(t_{n-1}^{(s)} + r_{n-1}^{(s)} v_{n-1}^{(s)}; \sigma_t^2). \tag{5}$$

The measured values of the parameters are $\sigma_t = 0.02$ s and $\lambda = 0.0101$ s [15] (the value of σ_v will be explained later). Remarks should be made here: First, the number of observed onsets must be N_s+1 so that there are N_s IOIs corresponding to N_s score notes. Second, we do not put a distribution on the onset time of the first note $t_1^{(s)}$ because we formulate the model to be invariant under time translations and this value would not affect any results of inference. We will use the notation $\boldsymbol{v}^{(s)} = (v_n^{(s)})_{n=1}^{N_s}$ and $\boldsymbol{t}^{(s)} = (t_n^{(s)})_{n=1}^{N_s+1}$.

Finally we describe the generation of pitches $\boldsymbol{p}^{(s)} = (p_n^{(s)})_{n=0}^{N_s+1}$ as a Markov chain (we introduce an auxiliary symbol $p_0^{(s)}$ for later convenience). The probabilities are

$$p_1^{(s)} | p_0^{(s)} \sim \mathsf{Cat}(\boldsymbol{\theta}_{p_0^{(s)}}^{(s)}), \tag{6}$$

$$p_n^{(s)} | p_{n-1}^{(s)} \sim \mathsf{Cat}(\boldsymbol{\theta}_{p_{n-1}^{(s)}}^{(s)}) \quad (n = 2, \ldots, N_s+1), \tag{7}$$

where $\boldsymbol{\theta}_{p_0^{(s)}}^{(s)} = (\theta_{p_0^{(s)},p}^{(s)})_p$ is the initial probability, and $\boldsymbol{\theta}_{p_{n-1}^{(s)}}^{(s)} = (\theta_{p_{n-1}^{(s)},p}^{(s)})_p$ is the (stationary) transition probability. These parameters are to be learned from music data.

The above model can be summarised as an autoregressive HMM, which we call a voice HMM, with hidden states $(\boldsymbol{r}^{(s)}, \boldsymbol{v}^{(s)})$ and outputs $(\boldsymbol{p}^{(s)}, \boldsymbol{t}^{(s)})$. Although so far the probabilities of pitches are independent of other variables, they will be significant once multiple voice HMMs are merged and the posterior probabilities are inferred.

3.2 Model for Multiple Voices

We combine the multiple voice HMMs in Sec. 3.1 using the framework of merged-output HMMs [19]. Simply speaking, the sequence of merged outputs is obtained by gathering the outputs of the voice HMMs and sorting them according to onset times. To derive inference algorithms that are computationally tractable, however, we should formulate a model that outputs notes incrementally in the order of observations. This can be done by introducing stochastic variables $\boldsymbol{s} = (s_n)_{n=1}^{N+1}$, which indicate that the n-th observed note belongs to voice s_n, with the following probability:

$$s_n \sim \mathsf{Ber}(\alpha_1, \alpha_2), \tag{8}$$

where Ber is the Bernoulli distribution. α_{s_n} represents how likely the n-th note is generated from the HMM of voice s_n and, to improve the results of voice separation, we put on the parameter conditional dependence on the lowest and highest pitches of simultaneously sounding notes.

If voice s_n is chosen, then the HMM of voice s_n outputs a note, and the hidden state of the other voice HMM is unchanged. Such a model can be described with an HMM with a state space labelled by $k_n = (s_n, p_n^{(1)}, r_n^{(1)}, t_n^{(1)}, p_n^{(2)}, r_n^{(2)}, t_n^{(2)}, v_n)$. Here we have a single tempo variable v_n that is shared by the two voices in order to assure loose synchrony between them. $P(k_n | k_{n-1})$,

for $n \geq 2$, is given as

$$\alpha_{s_n} P(v_n|v_{n-1}) A^{(s_n)}_{r^{(s_n)}_{n-1} r^{(s_n)}_n}(p^{(s_n)}_n, t^{(s_n)}_n | p^{(s_n)}_{n-1}, t^{(s_n)}_{n-1}; v_{n-1}) \cdot \Big[\delta_{s_n 1} \delta_{r^{(2)}_{n-1} r^{(2)}_n} \delta_{p^{(2)}_{n-1} p^{(2)}_n} \delta(t^{(2)}_{n-1} - t^{(2)}_n) + (1 \leftrightarrow 2)\Big], \quad (9)$$

where we have defined

$$A^{(s)}_{r^{(s)}_{n-1} r^{(s)}_n}(p^{(s)}_n, t^{(s)}_n | p^{(s)}_{n-1}, t^{(s)}_{n-1}; v_{n-1}) = \pi^{(s)}_{r^{(s)}_{n-1}, r^{(s)}_n} \theta^{(s)}_{p^{(s)}_{n-1}, p^{(s)}_n} P(t^{(s)}_n | t^{(s)}_{n-1}, v_{n-1}, r^{(s)}_{n-1}) \quad (10)$$

and δ denotes Kronecker's delta for discrete variables and Dirac's delta function for continuous variables. The probability $P(v_n|v_{n-1})$ is defined in Eq. (3), and $P(t^{(s_n)}_n | t^{(s_n)}_{n-1}, v_n, r^{(s_n)}_n)$ is defined in Eqs. (4) and (5). For note values the initial probability is given as $r^{(s)}_1 \sim \mathsf{Cat}(\boldsymbol{\pi}^{(s)}_{\mathrm{ini}})$, and for pitches the initial probability is set as in Eq. (6). The first onset times $t^{(1)}_1$ and $t^{(2)}_1$ do not have distributions, as explained in Sec. 3.1, and we practically set $t^{(1)}_1 = t^{(2)}_1 = t_1$ where t_1 is the first observed onset time. Finally the output of the model is given as

$$p_n = p^{(s_n)}_n, \quad t_n = t^{(s_n)}_n, \quad (11)$$

and thus the complete-data probability is written as

$$P(\boldsymbol{k}, \boldsymbol{p}, \boldsymbol{t}) = \prod_n P(k_n|k_{n-1}) \delta_{p_n p^{(s_n)}_n} \delta(t_n - t^{(s_n)}_n). \quad (12)$$

$N = N_1 + N_2$ denotes the total number of score notes, and the following notations will be used: $\boldsymbol{v} = (v_n)^N_{n=1}$, $\boldsymbol{p} = (p_n)^{N+1}_{n=1}$, $\boldsymbol{t} = (t_n)^{N+1}_{n=1}$, and $\boldsymbol{k} = (k_n)^{N+1}_{n=1}$. Note that whereas $\boldsymbol{p}$ and $\boldsymbol{t}$ are observed quantities, $\boldsymbol{p}^{(1)}, \boldsymbol{p}^{(2)}, \boldsymbol{t}^{(1)}, \boldsymbol{t}^{(2)}$ are not because we cannot directly observe the voice information encrypted in $\boldsymbol{s}$.

3.3 Inference Algorithm

Rhythm transcription based on the proposed model can be performed by estimating the most probable hidden state sequence $\hat{\boldsymbol{k}}$ given the observations $(\boldsymbol{p}, \boldsymbol{t})$. Once $\hat{\boldsymbol{k}}$ is obtained, we can extract the voice information $\hat{\boldsymbol{s}}$ and the note values $\hat{\boldsymbol{r}}^{(1)}$ and $\hat{\boldsymbol{r}}^{(2)}$. These are the result of voice separation and rhythm transcription.

The maximisation of the probability $P(\boldsymbol{k}|\boldsymbol{p}, \boldsymbol{t})$ can be in principle done with the Viterbi algorithm [13]. However, due to the complexity of our model, we need refinements to the standard Viterbi algorithm to derive a computationally tractable algorithm. First, since the state space of the merged-output HMM in Sec. 3.2 involve both discrete and continuous variables, an exact inference is not computationally tractable. To solve this problem, we discretise the tempo variable in a range that is common in music practice. Other continuous variables $\boldsymbol{t}, \boldsymbol{t}^{(1)}$, and $\boldsymbol{t}^{(2)}$ can take only values of observed onset times and thus can, in effect, be treated as discrete variables.

Second, it appears that a Viterbi algorithm derived in the way proposed in [19] has rather large computational cost for the present model and in practice difficult to execute. The large computational cost derives from the fact that we need to model pitches and onset times for the voice HMMs. This problem can be reduced by noting that the pitch and onset time are observed quantities and can be represented by a variable describing the historical information of voices associated to notes, as suggested in [20]. Extending the formalism of introducing a latent variable to describe this information, we can derive an efficient algorithm. Details will be given in our forthcoming paper [23]. We have confirmed that this algorithm can be executed in a standard modern computer environment with a practical time (within a few hours for a performance with hundreds of notes).

4. EVALUATION

4.1 Setup

We evaluated the proposed model by comparing the accuracy of its rhythm transcription with that of previously studied models based on HMMs. Two data sets of MIDI recordings of classical piano pieces were used. One ('polyrhythmic' data set) consisted of 18 performances of 15 (excerpts of) pieces that contained 2 against 3 or 3 against 4 polyrhythmic passages, and the other ('standard polyphony' data set) consisted of 30 performances of 22 pieces that did not contain polyrhythmic passages. Pieces by various composers, ranging from J. S. Bach to Debussy, were chosen and the players were also various: Some of the performances were taken from the PEDB database [22], a few were performances we recorded, and the rest was taken from public domain websites.

All normal, dotted, and triplet note values ranging from the whole note to the 32nd note were used as candidate note values. The transition and initial probabilities of the note values and pitches, and the value of α_s, were learned from a data set of classical piano scores that had no overlap with the test data. For the tempo variable, we discretised v_n into 50 values logarithmically equally spaced in the range of 0.3 to 1.5 sec per quarter note (corresponding to 200 BPM and 40 BPM). The standard deviation in Eq. (3) was set as $\sigma_v = 1.08$, using the value in [15] as a reference.

For comparison, we implemented the note HMM [6] and the metrical HMM [8] that is extended to handle polyphony. The parameters of the score models were also trained with the same score dataset. The performance model was the same as that for the proposed model.

We used as an evaluation measure the rhythm correction ratio, *i.e.*, the ratio of the smallest number of edit operations needed to correct the estimated result to the number of notes in the data. In addition to note-wise correction (shift operation), the scaling operation applied for a subsequence of note values was included. This is because there is arbitrariness in choosing the unit of note values: For example, a quarter note played in a tempo of 60 BPM has the same duration as a half note played in a tempo of 120 BPM. The smallest number of necessary edit operations N_{e} can be calculated by a dynamic programming similar to that used in computation of the Levenshtein distance (see our forthcoming paper [23] for details). The rhythm correction ratio $\mathscr{R}$ is then given as $\mathscr{R} = N_{\mathrm{e}}/N$. When separated voices are given, we can apply the above editing of note values for each voice and then the total rhythm correction cost is the sum of the rhythm correction costs in all voices.

Data set	Model	$\mathscr{R}$ [%]
Polyrhythmic	Proposed	$\mathbf{16.0 \pm 3.6}$
	Note HMM [6]	28.9 ± 4.9
	Metrical HMM [8]	34.1 ± 5.0
Standard polyphony	Proposed	7.9 ± 1.3
	Note HMM [6]	$\mathbf{7.0 \pm 1.3}$
	Metrical HMM [8]	7.9 ± 1.4

Table 1. Average rhythm correction rates $\mathscr{R}$ with standard errors. Lower is better.

4.2 Results

Results in Table 1 show that the proposed model clearly outperformed the other models for performances with polyphonic passages. Fig. 4 shows an example that a polyrhythmic passage is successfully transcribed with the proposed model with minor errors[2]. We see that the proposed model correctly recognised the 3 against 4 polyrhythms. On the contrary, the Note HMM did not recognise the polyrhythms (cf. Fig. 2) and had frequent errors in chord clustering.

For performances in standard polyphony, on the other hand, the note HMM was slightly better than the proposed model and the metrical HMM. Presumably, the main reason is that the rhythmic pattern in the reduced sequence of chords is often simpler than that of melody/chords in each voice in the case of standard polyphony because of the principle of complementary rhythm [24]. In particular, notes/chords in a voice can have tied note values that are not contained in our candidate list (*e.g.* quarter note + 16th note value), which can also appear as a result of incorrect voice separation (Fig. 5). It is also observed that the transcription by the merged-output HMM can produce desynchronised cumulative note values in different voices. This is due to the lack of constraints to assure the matching of these cumulative note values and the simplification of independent voice HMMs. Further improvements are expected by incorporating such constraints and interactions between voices into the model.

For the note HMM and the proposed model, there were grammatically wrong sequences of note values, for example, triplets that appear in single or two notes without completing a unit of beat. This can be avoided with a refined score model with beat/bar structure [6, 11]. On the other hand, these grammatical errors were not observed in the transcriptions by the metrical HMM owing to the explicitly included metrical structure.

5. CONCLUSION

To develop a rhythm transcription algorithm that captures the voice structure, we constructed a stochastic model of musical score and performance using the framework of merged-output HMMs. The evaluation results confirmed that the proposed algorithm worked better for polyrhythmic performances than the previously proposed HMM-based algorithms.

[2] Sound files and more examples are accessible in our demonstration web page: `http://anonymous4721029.github.io/demo.html`

An important future direction of developing advanced transcription techniques is to capture the phrase or motivic structure of music. Recognition of offsets and articulations and detection of ornaments are challenging problems. The treatment of voice structure is a fundamental problem for these issues, and the results of this study may be applicable to solving these problems.

Acknowledgments

This work is partially supported by JSPS KAKENHI Nos. 24220006, 26240025, 26280089, 26700020, 15K16054, 16H01744 and 16J05486, JST OngaCREST Project and Kayamori Foundation. E. Nakamura is supported by the JSPS fellowship program.

6. REFERENCES

[1] A. Klapuri and M. Davy (eds.), *Signal Processing Methods for Music Transcription*, Springer, 2006.

[2] E. Benetos *et al.*, "Automatic Music Transcription: Challenges and Future Directions," *J. Intelligent Information Systems*, vol. 41, no. 3, pp. 407–434, 2013.

[3] H. Longuet-Higgins, *Mental Processes: Studies in Cognitive Science*, MIT Press, 1987.

[4] P. Desain and H. Honing, "The Quantization of Musical Time: A Connectionist Approach," *Comp. Mus. J.*, vol. 13, no. 3, pp. 56–66, 1989.

[5] T. Otsuki *et al.*, "Musical Rhythm Recognition Using Hidden Markov Model (in Japanese)," *J. Information Processing Society of Japan*, vol. 43, no. 2, pp. 245–255, 2002.

[6] H. Takeda *et al.*, "Hidden Markov Model for Automatic Transcription of MIDI Signals," *Proc. MMSP*, pp. 428–431, 2002.

[7] H. Takeda *et al.*, "Rhythm and Tempo Analysis Toward Automatic Music Transcription," *Proc. ICASSP*, vol. 4, pp. 1317–1320, 2007.

[8] C. Raphael, "Automated Rhythm Transcription," *Proc. ISMIR*, pp. 99–107, 2001.

[9] M. Hamanaka *et al.*, "A Learning-Based Quantization: Unsupervised Estimation of the Model Parameters," *Proc. ICMC*, pp. 369–372, 2003.

[10] A. Cemgil and B. Kappen, "Monte Carlo Methods for Tempo Tracking and Rhythm Quantization," *J. Artificial Intelligence Res.*, vol. 18 no. 1, pp. 45–81, 2003.

[11] M. Tsuchiya *et al.*, "Probabilistic Model of Two-Dimensional Rhythm Tree Structure Representation for Automatic Transcription of Polyphonic MIDI Signals," *Proc. APSIPA*, pp. 1–6, 2013.

[12] N. Takamune *et al.*, "Automatic Transcription from MIDI Signals of Music Performance Using 2-Dimensional LR Parser (in Japanese)," *Tech. Rep. SIGMUS*, vol. 2014-MUS-104, no. 7, pp. 1–6, 2014.

[13] L. Rabiner, "A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition," *Proc. IEEE*, vol. 77, no. 2, pp. 257–286, 1989.

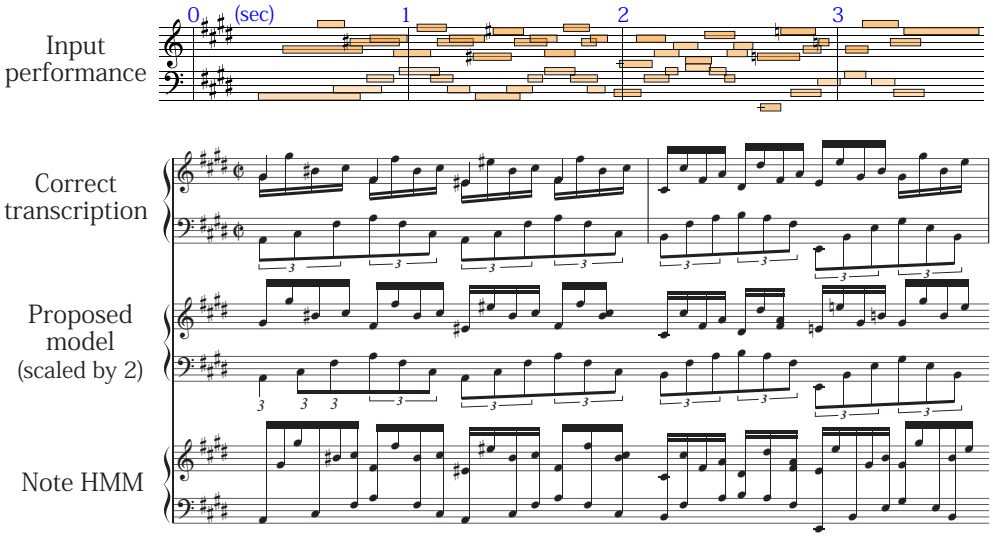


Figure 4. Transcription results of a polyrhythmic passage. For the result with the proposed model (merged-output HMM), the staffs indicate the estimated voices.

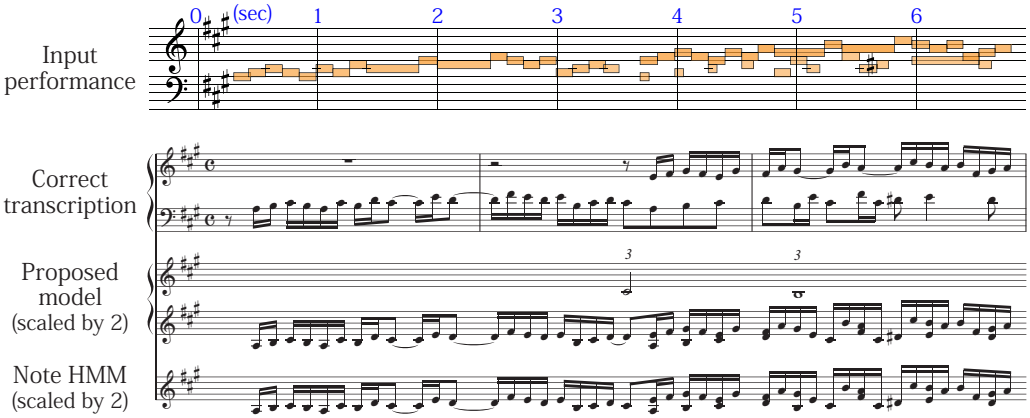


Figure 5. Transcription results of a standard polyphonic passage. For the result with the proposed model (merged-output HMM), the staffs indicate the estimated voices.

[14] C. Raphael, "Automatic Segmentation of Acoustic Musical Signals Using Hidden Markov Models," *IEEE Trans. on PAMI*, vol. 21, no. 4, pp. 360–370, 1999.

[15] E. Nakamura *et al.*, "A Stochastic Temporal Model of Polyphonic MIDI Performance with Ornaments," *J. New Music Res.*, vol. 44, no. 4, pp. 287–304, 2015.

[16] A. Cont, "A Coupled Duration-Focused Architecture for Realtime Music to Score Alignment," *IEEE Trans. on PAMI*, vol. 32, no. 6, pp. 974–987, 2010.

[17] H. Heijink *et al.*, "Data Processing in Music Performance Research: Using Structural Information to Improve Score-Performance Matching," *Behavior Research Methods, Instruments, & Computers*, vol. 32, no. 4, pp. 546–554, 2000.

[18] B. Gingras and S. McAdams, "Improved Score-Performance Matching Using Both Structural and Temporal Information from MIDI Recordings," *J. New Music Res.*, vol. 40, no. 1, pp. 43–57, 2011.

[19] E. Nakamura *et al.*, "Merged-Output Hidden Markov Model for Score Following of MIDI Performance with Ornaments, Desynchronized Voices, Repeats and Skips," *Proc. Joint ICMC|SMC 2014*, pp. 1185–1192, 2014.

[20] E. Nakamura *et al.*, "Merged-Output HMM for Piano Fingering of Both Hands," *Proc. ISMIR*, pp. 531–536, 2014.

[21] Z. Ghahramani and M. Jordan, "Factorial Hidden Markov Models," *Machine Learning*, vol. 29, pp. 245–273, 1997.

[22] M. Hashida *et al.*, "A New Music Database Describing Deviation Information of Performance Expressions," *Proc. ISMIR*, pp. 489–494, 2008.

[23] E. Nakamura *et al.*, in preparation.

[24] F. Salzer and C. Schachter, *Counterpoint in Composition: The Study of Voice Leading*, Columbia University Press, 1989.

LYRICLISTPLAYER: A CONSECUTIVE-QUERY-BY-PLAYBACK INTERFACE FOR RETRIEVING SIMILAR WORD SEQUENCES FROM DIFFERENT SONG LYRICS

Tomoyasu Nakano **Masataka Goto**
National Institute of Advanced Industrial Science and Technology (AIST), Japan
{t.nakano, m.goto}[at]aist.go.jp

ABSTRACT

This paper presents *LyricListPlayer*, a music playback interface for an intersong navigation and browsing that enables a set of musical pieces to be played back by music zapping based on lyrics words. In other words, this paper proposes a novel concept we call *consecutive-query-by-playback*, which is for retrieving similar word sequences during music playback by using lyrics words as candidate queries. Lyrics can be used to retrieve musical pieces from the perspectives of the meaning and the visual scene of the song. A user of LyricListPlayer can see time-synchronized lyrics while listening, can see word sequences of other songs similar to the sequence currently being sung, and can jump to and listen to one of the similar sequences. Although there are some systems for music playback and retrieval that use lyrics text or time-synchronized lyrics and there is an interface generating lyrics animation by using kinetic typography, LyricListPlayer provides a new style of music playback with lyrics navigation based on the local similarity of lyrics.

1. INTRODUCTION

Since a song's lyrics can be used to convey emotions/passions/feelings/thoughts and to facilitate imagine visual scenes, they are an important element helping listeners have emotional involvement to the song. In fact, some listeners are aware of lyrics while listening to music and use them as a criterion for selecting musical pieces [1].[1] Lyrics are text-based information and can be used as a retrieval query by music professionals and casual listeners. Indeed, there are many works focusing on how to retrieve/browse music by using lyrics [2–7].

Previous works investigating the use of lyrics in music information retrieval have focused on the following three approaches

- 1) keyword-based retrieval – Retrieving lyrics by using text-based keywords of music search web sites[2] . Retrieves based on a full-text search using lyrics text or metadata such as song titles are available, as are retrieves based on a released year or a decade, music genre, scene (supportive, love, spring, or summer), ranking, and comments from listeners. SyncPlayer [3], a query-by-lyrics retrieval system, can navigate from a list of retrieved results to the corresponding matching positions within the audio recordings.
- 2) content-based retrieval (song-level lyrics similarity/classification) – Retrieving/browsing lyrics by favorite lyrics via query-by-example systems. Lyrics can be used to retrieve songs by visualizing music archives [4, 6, 7] and recommended songs [8]. Automatic topic detection [2, 7, 9, 10] and semantic analysis [11] of song lyrics have also been proposed. Several approaches analyzed the text of lyrics by using natural language processing to classify lyrics according to emotions, moods, and genres [12–15].
- 3) hyperlinking lyrics [5] – Creating a hyperlink from a word sequence in the lyrics of a song to the same sequence in the lyrics of another song and using the hyperlink for navigating/discovering lyrics.

We propose a music playback interface, *LyricListPlayer*, that is based on an extension of a hyperlinking lyrics system [5]. The paper describing that the previous system focused on creating keyword-based hyperlinks without interaction and just mentioned using the hyperlinking structure as a basis for imaging applications. In contrast, we focused on creating similarity-based hyperlinks with interaction to

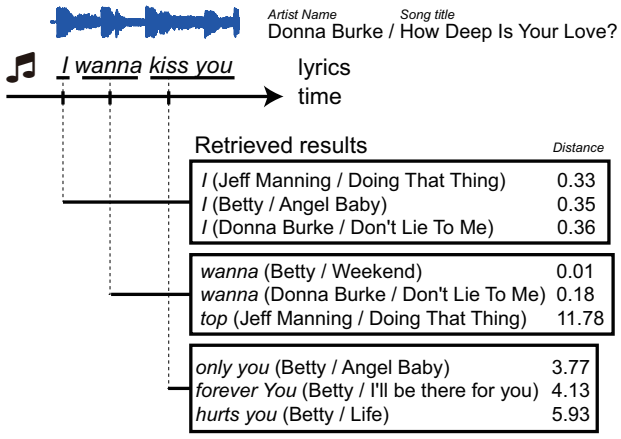


Figure 1. Consecutive-query-by-playback: LyricListPlayer uses lyrics (word sequences) currently being sung as candidate queries, and similar word sequences from different song lyrics are immediately updated during music playback.

[1] In their questionnaire investigation, 66 of 86 subjects said they are usually conscious of lyrics while listening to music, and 42 of 86 subjects said they often choose songs based on lyrics [1].

[2] *e.g.*, MusiXmatch https://www.musixmatch.com/

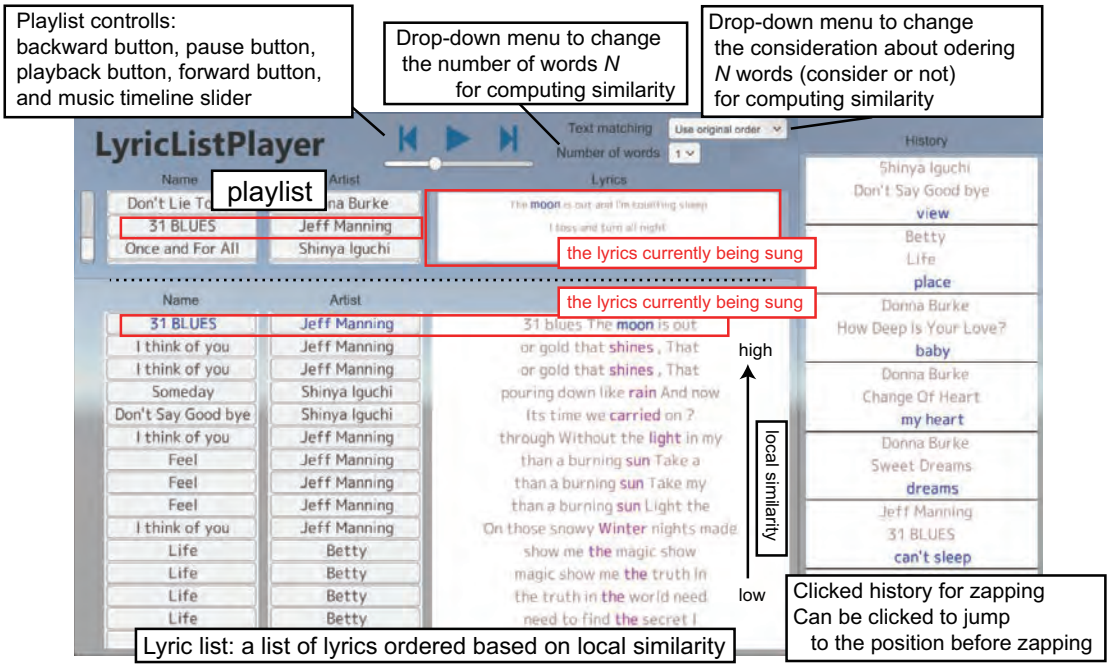


Figure 2. An example LyricListPlayer screen. When the query (a word currently being sung) is "moon", the interface can retrieve "shines," "rain," "light," and "sun."

increase a potential of the system. LyricListPlayer computes word-sequence-level similarity to cover linking not only identical word sequences but also similar sequences. It also increases the flexibility of retrieval by letting users change the length (number of words) and order of the sequence used for computing similarity.

Thus, LyricListPlayer has a potential to provide a new immersive style of music playback on the *Music Web* where songs are hyperlinked to each other on the basis of their lyrics [5]. Lyrics (word sequences) currently being sung can be issued as candidate queries automatically, and retrieved results are immediately updated during music playback. We call this novel concept of music information retrieval, consecutive-query-by-playback (Fig. 1).

The interface displays time-synchronized lyrics and uses lyrics (words) of a song currently being played back as a query. LyricListPlayer can also retrieve local similar lyrics and they can be played back to check sung style (vocal timbre and melody) and/or sung context (story of the lyrics). Similar lyrics, which like the currently sung lyrics are changed from moment to moment, are also displayed and they can be clicked to listen to them immediately. To compute lyrics similarity, latent meanings (topics) behind the words are estimated. The interface can retrieve words that are in some way similar to a query word. When the query is "angel", for example, the interface can retrieve "snuggle" and "love."

2. LYRICLISTPLAYER: AN INTERFACE FOR QUERY CANDIDATES GENERATION BY MUSIC PLAYBACK

LyricListPlayer is a music playback interface for a set of songs, and similar word sequences from the song currently being played back are displayed. Interaction and hyperlinked relationships between songs can provide a new perspective as a combination of a passive music retrieving interface and an active music listening interface [16], a combination with which a user can browse and discover songs by just listening to music and clicking a similar word sequence to jump to listen from there.

Figure 2 shows the LyricListPlayer screen. A music playlist is shown at top of the figure. The interface displays not only the lyrics of the song currently played back song (Fig. 2 top) but also its similar lyrics list (Fig. 2 bottom). The top of the list shows the lyrics of the song currently played back, and the other listed lyrics are ordered based on local similarity of latent topics. The list is called a "lyric list" in this paper. Hereafter, all lyrics in screenshots illustrated in this paper are taken from the RWC Music Database (Popular Music) [17]. Twenty songs (RWC-MDB-P Nos.81–100) are used as a playlist and the lyric list is also estimated from the songs.

LyricListPlayer provides the following three functions.

1) display music-synchronized lyrics
2) display music-synchronized lyric list
3) lyrics zapping interaction

2.1 Display music-synchronized lyrics and lyric list (similar word sequences)

The word currently being sung is highlighted by blue coloring so that, as in karaoke, a user can easily follow the current playback position. As its retrieved results, the similar word sequences are colored magenta (Fig. 3).

The range of coloring is determined by the length of the word sequence N (the number of morphemes or words) considered for computing local similarity. The local similarity of the lyrics of the lyric list is changed based on N and whether or not the ordering is considered. For exam-

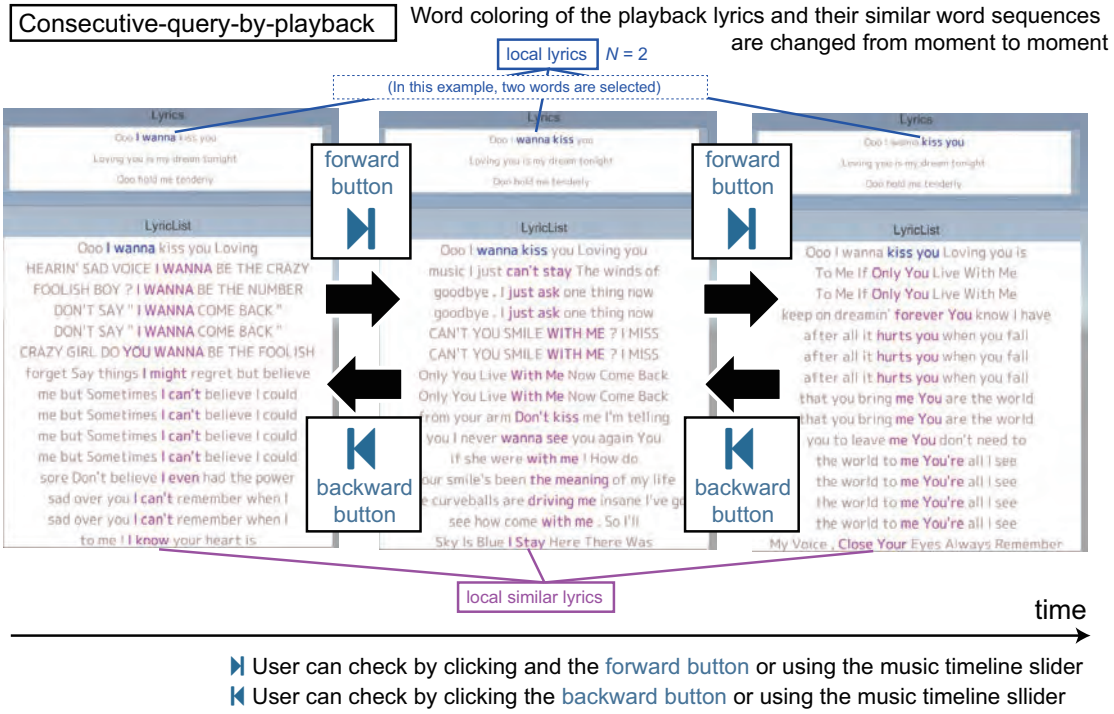


Figure 3. An example of changing playback position and similar word sequences along with the playback word.

ple, if the ordering is not considered, the word sequence "A B C" in a song has the same similarity as the sequence "A C B" in the same song.[3] Figure 4 and 5 show similar word sequences with different conditions: different N and with or without consideration of the words' ordering.

The coloring design is different from that used in the interface LyricSynchronizer and those used in well-known karaoke systems. LyricListPlayer colors the word (morpheme) currently sung and the subsequent $N-1$ words. The reason for this coloring design is to show information about both the current playback position' and the length currently used to compute local similarity'. To show the context of the similar word sequences in the lyric list, the previous and next words are also displayed.

The length of the word sequence (the number of words) and whether or not ordering is considered in computing similarity can be changed by using the two drop-down menus at the top of the screen (Fig. 2 top). The current playback position can be changed by using the forward and backward buttons to jump to the next/previous word (morpheme) or by using the music timeline slider.

2.2 Lyrics zapping interaction

The user can show the sung lyrics and their similar word sequences while listening to music. This is a kind of *passive* music information retrieval because the user does not input a query explicitly/actively.

On the other hand, the displayed similar word sequences can be clicked for zapping, to jump to listen from that point (Fig. 6). The zapping history is displayed at the right-side of the screen (Fig. 2) and can be clicked to back to a song played back before zapping. In addition, as a potential of the lyric list, the similar word sequences can be played back continuously to get an overview of the sequences sung by different artists or contexts.

[3] In the current implementation, the same word sequences of different songs have different similarities.

3. IMPLEMENTATION

The system first synchronizes the phoneme-level pronunciation of the lyrics with the musical audio signals for all the songs in the playlist. The estimated onset time and durations of all phonemes are converted to morpheme-level for Japanese lyrics and to word-level for English lyrics. This synchronization is called *lyrics alignment*.

Then, to compute similarity between words, the system estimates the latent topics of lyrics. Finally, the system calculates similarity among all word sequences with different N in the range $N = 1, 2, ..., 5$. The indexes of 200 word sequences having high similarity for each word sequence are stored for display on the lyric list screen. This interface support Japanese and English lyrics, and Japanese lyrics are spelled in a mixture of Japanese phonetic characters and Chinese characters.

3.1 Lyrics alignment

The phonetic-to-audio synchronization is estimated through Viterbi alignment with a phoneme-level hidden Markov model (monophone HMM) that is used as an acoustic model. We trained Japanese and English monophone HMMs by using the RWC Music Database (Popular Music) [17] with our own phonetic annotations; 80 Japanese songs are used to train a Japanese acoustic model and 20 English songs are used to train an English acoustic model. Here we refer this song set as the RWC

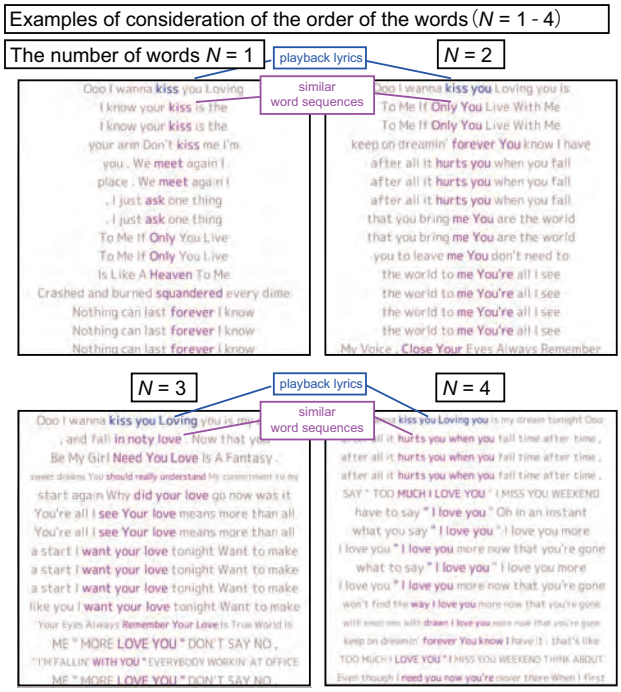


Figure 4. Examples of different N, the length of the word sequence (number of morphemes or words) for computing local similarity, with consideration of the ordering of the words.

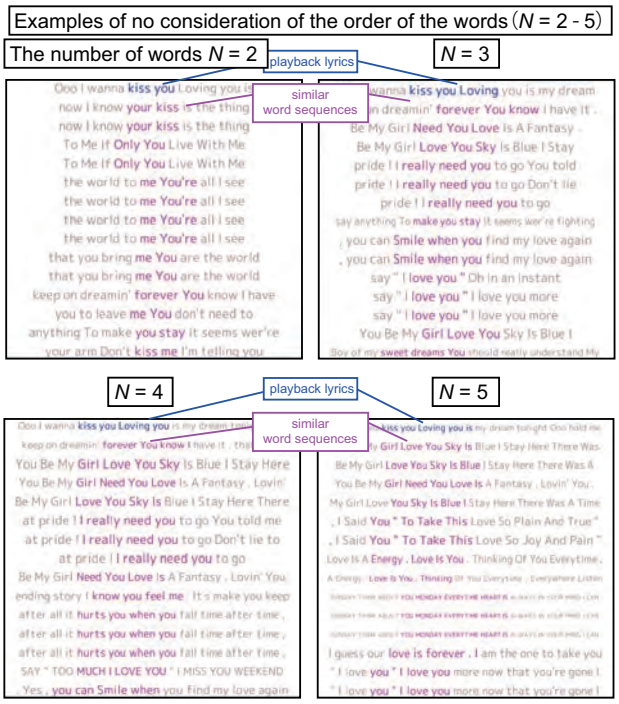


Figure 5. Examples by different N, the length of the word sequence (number of morphemes or words) for computing local similarity, with no consideration of the ordering of the words.

MDB.

To train the acoustic models, the pronunciation is estimated by using the Japanese language morphological analyzer MeCab [18] and the CMU pronouncing dictionary[4] for English lyrics. The acoustic features and alignment method are based on those used in LyricSynchronizer [19]. With regard to the acoustic features, we target monaural 16-kHz digital recordings and extract Δpower, 12th-order MFCCs, and 12th-order ΔMFCCs every 10 ms. To estimate the features, we performed separation of vocals from polyphonic musical audio signals [19].

3.2 Topic modeling

We use latent Dirichlet allocation (LDA) [20] for lyrics topic modeling. Since the LDA was originally proposed for text analysis, it can be used for lyrics modeling. In fact, there are three papers on work that used lyrics for LDA-based music retrieval [2, 7, 10]. The number of topics K is set to 100, and the model parameters of LDA are trained using the collapsed Gibbs sampler [21]. The conditions are based on previous work [7, 22].

The song set used for Japanese model training is 1,896 Japanese popular songs[5] and 80 lyrics of the RWC MDB. The Japanese popular songs appeared on a popular music chart in Japan[6] and were placed in the top twenty on weekly charts appearing between 2000 and 2008. The song set used for English model training is 2,314 English songs sung by 2,314 artists from Music Lyrics Database v.1.2.7[7], 51 English songs from commercial music CDs and 61 English lyrics of the RWC Music Database (20 from Popular Music, 10 from Royalty-Free Music, and 31 from Music Genre) [17, 23] are used.

For the topic modeling, all morphemes of Japanese are converted to the original form by using the MeCab for Japanese lyrics. Symbols such as punctuation marks and exclamation marks are used for model training because they can be used to express emotions or feelings. Finally, the vocabulary size in the 1,976 Japanese lyrics is 19,390 words (morphemes), and the vocabulary size in the 2,426 English lyrics is 23,756 words.

3.3 Similarity computing

By using a variational Bayesian inference of the LDA model training, the *responsibility*[8] (mixing weights) of multiple topics for each word can be estimated. Then the responsibilities of a word can be interpreted as the number of observations of the corresponding topic. To obtain responsibilities (unigram probabilities) for a set of words with the length N of 2 or more without consideration of the ordering, the word's responsibilities are summed (Fig. 7). Since this summing approach can be used to compute similarity between two sets of words with different N, it can also be used to compute similarity between two lines.

Since each topic can be represented by a unigram probability of the vocabulary, the distance between two words

[4] http://www.speech.cs.cmu.edu/cgi-bin/cmudict

[5] Note that some are Western popular songs and English is used in them.

[6] http://www.oricon.co.jp/

[7] http://www.odditysoftware.com/page-datasales1.htm

[8] This term is from an article [24].

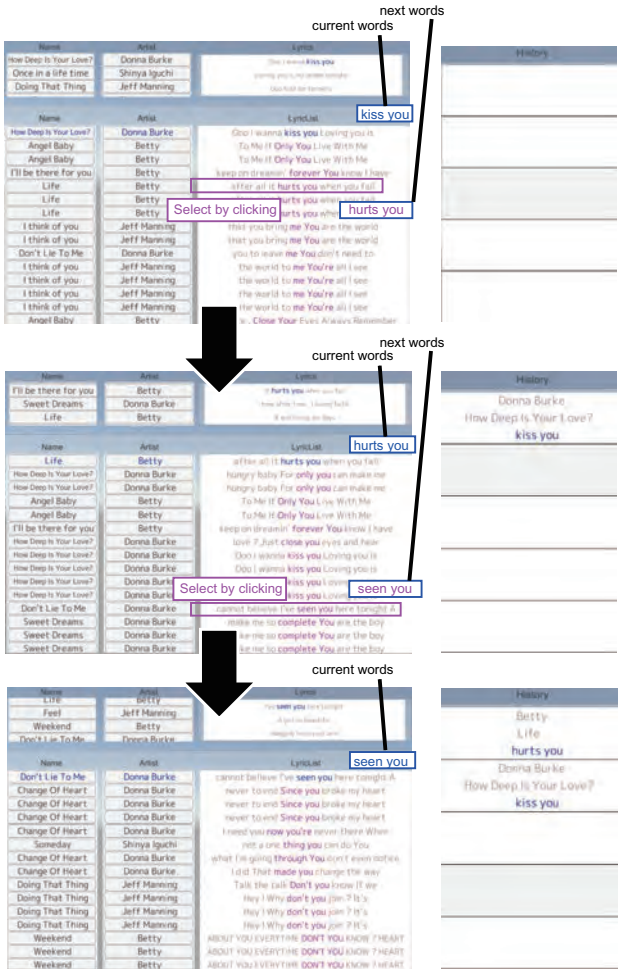


Figure 6. The local similar word sequences can be clicked to jump to listen from that point (lyrics zapping interaction).

is defined in this paper as the symmetric Kullback-Leibler distance (KL2) between two unigram probabilities.

To calculate similarity for words with the length N of 2 or more with consideration of the ordering, we first compute each word-pair similarity. In the current implementation, their median value is used as the similarity. For example, for two set of words, ABC and DEF, similarities A-D, B-E, and C-F are calculated first.

4. USER FEEDBACK

To investigate the capabilities, limitations, and potential of our interaction design, we asked eight users to use the system for 20 minutes and collected preliminary user feedback. We chose users, seven males and one female (U1–U8), who had different types of appreciation of music with lyrics. Five users had been conscious of lyrics while listening to music (U1, U2, U4, U5, and U8). Additionally, two user had occasionally chosen songs based on lyrics (U4 and U8).

The playlist consisted of the 10 Japanese songs, and we used the Japanese lyrics topic model. All users knew more than 1 song and six users knew 4 songs or more (U1, U2, U3, U4, U5, and U6). After the trial usage, we asked the

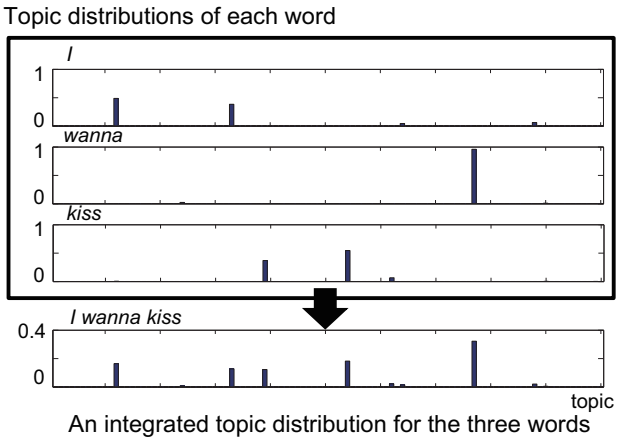


Figure 7. Examples of topic distributions and their integration.

users to write comments about two of the three primary functions of our system: 1) lyrics display (similar word sequences) and 2) lyrics zapping interaction.

Positive comments about the capabilities and potential of the interface were obtained from the users. Three users (U3, U4, and U7) used the function to display and listen to similar word sequences with enjoyment, and two users (U1 and U5) indicated that the function is helpful/useful for retrieving lyrics. One user (U6) indicated that the function changing the length of words for computing similarity was easy to use. One user (U3) frequently changed the length ($N = \{3, 4, 5\}$), and another user (U2) used only $N = 1$. With respect to the zapping interaction, three users (U2, U4 and U6) felt good about jumping/listening to the corresponding matching positions within the audio recordings.

Three users (U2, U6, and U7) indicated that LyricListPlayer is useful to be conscious of lyrics while listening to music. Furthermore, five users (U2, U5, U6, U7, and U8) indicated that a function zapping automatically is a way to expand possibility of application.

On the other hand, all users indicated that the speed with which the similar word sequences change should be controlled or be more effective and that either controlling it or making it more effective would be a good future direction for interface improvement. In association with that, four users (U2, U4, U6, and U8) thought that uncharacteristic words (*e.g.*, prepositions) should be omitted from the retrieving process. To improve practical utility of the system, five users (U2, U5, U6, U7, and U8) wanted to know a similarity between two songs based on lyrics or acoustic features (*e.g.*, mood, melody and musical structure).

5. CONCLUSION

This paper presents LyricListPlayer, a lyrics-synchronized music playback interface for retrieving lyrics passively. LyricListPlayer is also an active music listening interface based on lyrics, and an active listening style could help people be conscious of lyrics while listening to music. By taking into account the meaning of lyrics while listening, listeners can enrich their listening experience and become more emotionally involved with songs.

LyricListPlayer has an interactive function change the word length used for computing local similarity. Although there are works focusing on the similarity of music fragments or entire musical pieces [25] and the use of similarity by DJs connecting two pieces smoothly and for musical browsing [26, 27], to our knowledge, there is no research on how an interaction could be used to change the local range. Since music is a time-series media content, local similarity is an important aspect to deal with.

In future work, we plan to consider various word lengths for computing local similarity. We are also going to explore interactive designs for the display of similar word sequences, that is, to improve the interaction in ways based on user feedback. Although this interface focused only on lyrics-based information, information about other musical elements, such as vocal timbre and melody, should be integrated to enrich user experience. Moreover, a framework that can deal with a large amount of songs is also important to music listeners.

LyricListPlayer focused on an interaction design to explore "how to listen to a set of songs by using lyrics". The digitization of music and the distribution of content over the web have greatly increased the number of musical pieces available. Although music recommender systems and music information retrieval methods facilitate retrieving and listening to a large set of music, a recommended set of songs have to be listened to determine which song is one's favorite. Since the time one can spend listening to music is limited, more investigations of interactions for listening to a musical piece and/or a set of pieces are needed.

Acknowledgments

This paper utilized the RWC Music Database (Popular Music, Royalty-Free Music, and Music Genre). This work was supported in part by CREST, JST.

6. REFERENCES

[1] W. Machida and T. Itoh, "Lyricon: A visual music selection interface featuring multiple icons," in *Proc. IV 2011*, 2011, pp. 145–150.

[2] E. Brochu and N. de Freitas, ""Name That Song!": A probabilistic approach to querying on music and text," in *Proc. of NIPS2002*, 2002, pp. 1505–1512.

[3] M. Müller *et al.*, "Lyrics-based audio retrieval and multimodal navigation in music collections," in *Proc. ECDL'07*, 2007, pp. 112–123.

[4] R. Neumayer and A. Rauber, "Multi-modal music information retrieval – visualisation and evaluation of clusterings by both audio and lyrics," in *Proc. ISMIR 2007*, 2007.

[5] H. Fujihara *et al.*, "Hyperlinking Lyrics: A method for creating hyperlinks between phrases in song lyrics," in *Proc. ISMIR 2008*, 2008, pp. 281–286.

[6] D. Baur *et al.*, "SongWords: Exploring music collections through lyrics," in *Proc. ISMIR 2010*, 2010.

[7] S. Sasaki *et al.*, "LyricsRadar: A lyrics retrieval system based on latent topics of lyrics," in *Proc. ISMIR 2014*, 2014, pp. 585–590.

[8] R. Takahashi *et al.*, "Building and combining document and music spaces for music query-by-webpage system," in *Proc. of Interspeech 2008*, 2008, pp. 2020–2023.

[9] F. Kleedorfer, "Oh Oh Oh Whoah! towards automatic topic detection in song lyrics," in *Proc. ISMIR 2008*, 2008.

[10] L. Sterckx *et al.*, "Assessing quality of unsupervised topics in song lyrics," in *Advances in Information Retrieval, ECIR 2014*, 2014, pp. 547–552.

[11] B. Logan, "Semantic analysis of song lyrics," in *Proc. ICME 2004*, 2004.

[12] C. Laurier, "Multimodal music mood classification using audio and lyrics," in *Proc. ICMLA 2008*, 2008.

[13] C. McKay *et al.*, "Evaluating the genre classification performance of lyrical features relative to audio, symbolic and cultural features," in *Proc. ISMIR 2008*, 2008, pp. 213–218.

[14] D. J. Hu and L. K. Saul, "A probabilistic topic model for unsupervised learning of musical key-profiles," in *Proc. of ISMIR2009*, 2009, pp. 441–446.

[15] M. V. Zaanen and P. Kanters, "Automatic mood classification using TF*IDF based on lyrics," in *Proc. ISMIR 2010*, 2010, pp. 75–80.

[16] M. Goto, "Active music listening interfaces," in *Proc. ICASSP 2007*, 2007.

[17] M. Goto *et al.*, "RWC Music Database: Popular, classical, and jazz music databases," in *Proc. ISMIR 2002*, 2002, pp. 287–288.

[18] T. Kudo, "MeCab: Yet another part-of-speech and morphological analyzer," http://mecab.sourceforge.net/.

[19] H. Fujihara *et al.*, "LyricSynchronizer: Automatic synchronization system between musical audio signals and lyrics," *IEEE Journal of Selected Topics in Signal Processing*, vol. 5, no. 6, pp. 1252–1261, 2011.

[20] D. M. Blei *et al.*, "Latent Dirichlet allocation," *Journal of Machine Learning Research*, vol. 3, pp. 993–1022, 2003.

[21] T. L. Griffiths and M. Steyvers, "Finding scientific topics," in *Proc. Natl. Acad. Sci. USA (PNAS)*, vol. 1, 2004, pp. 5228–5235.

[22] T. Nakano *et al.*, "Musical similarity and commonness estimation based on probabilistic generative models," in *Proc. IEEE ISM 2015*, 2015.

[23] M. Goto *et al.*, "RWC Music Database: Music genre database and musical instrument sound database," in *Proc. of ISMIR 2003*, 2003, pp. 229–230.

[24] C. M. Bishop, *Pattern recognition and machine learning*. Springer-Verlag New York, Inc., 2006.

[25] M. Goto and K. Hirata, "Recent studies on music information processing," *Acoustical Science and Technology (edited by the Acoustical Society of Japan)*, vol. 25, no. 6, pp. 419–425, 2004.

[26] H. Ishizaki *et al.*, "Full-automatic DJ mixing with optimal tempo adjustment based on measurement function of user discomfort," in *Proc. ISMIR 2009*, 2009, pp. 135–140.

[27] T. Hirai *et al.*, "MusicMixer: Computer-aided DJ system based on an automatic song mixing," in *Proc. ACE 2015*, 2015, pp. 1–5.

LAZY EVALUATION IN MICROSOUND SYNTHESIS

Hiroki Nishino
Imagineering Institute, Malaysia &
Chang Gung University, Taiwan
hiroki.nishino@acm.org

Adrian David Cheok
Imagineering Institute, Malaysia &
City University London, United Kingdom
adrian@imagineeringinstitute.org

ABSTRACT

The microsound synthesis framework in the LC computer music programing language integrates objects and library functions that can directly represent microsounds and related manipulations for microsound synthesis. Together with the mechanism that enables seamless collaboration with the unit-generator-based sound synthesis framework, such abstraction can help provide a simpler and terser programing model for various microsound synthesis techniques.

However, while the microsound synthesis framework can achieve practical real-time sound synthesis performance in general, it was observed that temporal suspension in sound synthesis can occur, when a very large microsound object beyond microsound time-scale is manipulated, missing the deadline for real-time sound synthesis.

In this paper, we describe our solution to this problem. By lazily evaluating microsound objects, computation is delayed until when the samples are actually needed (e.g., for the DAC output), and, when performing the computation, only the amount of samples required at the point is computed; thus, temporal suspension in real-time sound synthesis can be avoided by distributing the computational cost among the DSP cycles. Such a solution is beneficial to extend the application domains of the sound synthesis framework design beyond microsound synthesis towards more general sound synthesis techniques.

1. INTRODUCTION

Today, microsound synthesis techniques [1] already constitute an important part of digital sound synthesis techniques for musical creation, being used for both non-real-time and real-time sound synthesis. Unlike many other sound synthesis techniques that conceptualize sounds as functions of time, microsound synthesis conceptualizes the sound as a composition of many short sound particles that overlap-add onto each other. Such a significant conceptual difference led to the question if the traditional unit-generator concept [2, p.89], which describes a sound synthesis algorithm by software modules that stream sample data to each other, is still appropriate also for microsound synthesis. While there have not been many examples, some previous works investigate more suitable software abstractions for microsound synthesis for this reason.

The LC computer music programming language [3] that we developed is one of the most recent examples of this kind, which takes microsound synthesis techniques into account in the software abstraction. The microsound synthesis framework in the LC language [4] significantly differs from existing unit-generator-based synthesis frameworks and integrates objects and library functions that can directly represent microsounds and related manipulations in microsound synthesis. Such abstraction contributes to describing various microsound synthesis techniques much simpler and terser in comparison with many existing unit-generator languages and can realize microsound synthesis in real-time.

Yet, as we previously described in [4], temporal suspension of real-time sound synthesis can be observed in certain situations; as LC's microsound objects are arrays of sample values (with useful methods) in nature, when manipulating a microsound object of a very large size, far beyond the microsound time-scale, the deadline for the real-time sound synthesis can be missed, causing temporal suspension of the audio output audible to human ears. We also described that this issue has normally not been observed in most practical situations when the sizes of microsound objects stay reasonably within the microsound time-scale and microsounds are manipulated sporadically as assumed by the sound synthesis framework. Yet, it is still desirable to avoid such temporal suspensions by large microsound objects to extend the application domain to more general applications beyond microsound synthesis.

In this paper, we propose a solution to this problem by adopting lazy evaluation to the microsound synthesis framework. Instead of eagerly evaluating the results of manipulations right when they are performed, the evaluation is delayed until when the result is actually required. Also, the evaluation takes place only for the number of samples required at that point, rather than for the entire microsound object; thus, by the adoption of lazy evaluation to the manipulation of microsound objects, the pause time can be significantly reduced and temporal suspension can be avoided in most practical situations, even when manipulating large microsound objects.

To assess the actual performance efficiency without being influenced by other factors in the language implementation (e.g., memory allocation, garbage-collection, or task-switching), we implemented a testing software framework and measured the performance efficiency in C++. The results showed a significant reduction of pause time, just as expected.

Such adoption of lazy evaluation to microsound objects, not only solves the issue of the temporal suspensions in most practical situations, but can also contribute to making the microsound synthesis framework more stable at runtime, by distributing the computational cost beyond one DSP cycle; this is quite favorable towards more general applications of the sound synthesis framework design, certainly beyond microsound synthesis.

2. RELATED WORK

2.1 Microsound Synthesis

Microsound synthesis was first brought to the practice of computer music around the early 1970s[1]. Since then, various sound synthesis techniques of the kind have been developed. This includes granular synthesis [5], formant wave-function (FOF) (from *function d'onde formantique)* [6], FOF synthesis [7], and waveset synthesis [8], all these belonging the family of microsound synthesis.

While many other synthesis techniques that conceptualize a sound as a function of time (such as additive synthesis, subtractive synthesis and FM synthesis), microsound synthesis conceptualizes sound quite differently. Generally speaking, in microsound synthesis, the entire sound output is composed of many short sound particles (i.e., microsounds) that overlap-add onto each other. The duration of such short sound particles extends from "the thread of timbre perception (several hundred micro seconds) up to the duration of a short sound object (~100 msec)" to "the boundary between the audio frequency range (approximately 20 Hz to 20 kHz) and the infrasonic frequency range (below 20 Hz)" [1, p.21].

In fact, Gabor, whose theory had a significant influence on the origins of microsound synthesis, already contrasted his theory to "the orthodox method of analysis … [which] starts with the assumption that the signal is a function (t) of time t" [9].

2.2 Software Frameworks for Microsound Synthesis

Such a theoretical difference as described earlier led to the question of whether the traditional unit-generator concept is appropriate for microsound synthesis[2].

Typically, in unit-generator languages, microsound synthesis techniques are normally realized by unit-generators, which encapsulate microsound synthesis techniques within, or by implementing microsounds as note-level sound objects and scheduling them with the user program.

Yet, some computer music researchers discuss that such implementations may not be appropriate for microsound synthesis. In [10], Brandt argues that "Music-N languages like Csound [11]," which are typical unit-generator languages, "are too limited" and not appropriate for FOF synthesis, since "the stream [of grains in FOF synthesis] is irregularly timed, and a grain is a sequence of samples". Brandt also puts forth that the unit-generator concept is a "black-box primitive" in computer music language in nature and is problematic as "if a desired operation is not present, and cannot be represented as a composition of primitives, it cannot be realized within the language" [12, pp.4-5].

In [4], we too maintained that the unit-generator concept may not be truly beneficial for end-users even when note-level objects and scheduling algorithms are utilized for the implementation of a certain microsound synthesis technique and provided as a library function, especially when considering creative explorations by users. Even though the complicated implementation is hidden inside the library function, if the user needs to alter the sound synthesis algorithm beyond what is provided by the function, the user has to modify the hidden implementation within the library function; such a situation is clearly not desirable to support the activities of exploratory understanding and exploratory design[3].

Based on such considerations, researchers and developers have been investigating alternative software abstractions that are suitable for microsound synthesis. For example, Bencina's object-oriented software framework design for a granular synthesizer includes the objects that directly represent grains (microsounds). Brandt's Chronic language is another example [12]. Chronic is a computer music language built upon the OCaml language [13]. Brandt proposed the 'temporary type constructor' concept, which introduces "a relation to a one-dimension axis, which we call time" to type constructors[4]. In doing so, for example, the audio stream of granular synthesis can be defined by the type of "*Sample vec event ivec.*"; each grain is represented by *Sample vec* (a vector of samples with finite size) and it is scheduled with a timestamp (*event*) in the infinite length stream of such events (*ivec*). With this sort of abstraction, Chronic can provide the direct access to low-level sample data, which does not normally exist in unit-

[1] For instance, one of the earliest examples of microsound synthesis was the implementation of ascynronous granular synthesis in MUSIC-V by Curtis Roads in 1974. Roads [1, p.110].

[2] If interested, see [2] for the detailed discussion by Nishino et al.

[3] Blackwell and Green list such activities as sketching; design of typography, software, etc.; other cases where the final product cannot be envisaged and has to be 'discovered' as the examples of exploratory design and discovering structure of algorithm, or discovering the basis of classification as the examples of exploratory understanding [13]

[4] "A type constructor builds complex types from simpler ones. For example, C has the "pointer to…" type constructor and we can write this as "α *pointer,"* where α is a free type variable which might be. for example, int" [12, p.7]

generator languages; this feature is beneficial for peforming various microsound synthesis techniques just within the language, without the help of 'native' modules written in C++.

However, Bencina's software framework is mainly designed for stand-alone synthesizer software and Brandt's Chronic language is a non-real-time computer music language and significant reconsideration is required to be adopted for real-time sound synthesis, because of its acausal behavior [12, p.77].

2.3 The Microsound Synthesis Framework in LC

In contrast, our LC language is designed with the sound synthesis framework deemed appropriate for real-time sound synthesis. Sharing the same interest with Bencina's granular synthesis framework and Brandt's Chronic language for investigating in alternative software design besides the traditional unit-generator concept, the microsound synthesis framework in the LC computer music language was designed with objects and library functions that can directly represent microsounds and related manipulations for microsound synthesis.

```
01 //create a new Sample object from the buf no. 0.
02 LoadSndFile(0 "/sound1.aif");
03 var snd = ReadBuf(0, 256::samp);
04
05 //create another by generating a window.
06 var win = GenWindow(512::samp, \hanning);
07
08 //create Samples objects by the method calls.
09 var grain    = snd->applyEnv(win);
10 var halfAmp  = snd->amplify (0.5);
11 var octup    = snd->resample(snd.size / 2);
12 var reversed = snd->reverse();
13
14 //convert a Samples obj to a SampleBuffer obj.
15 var sbuf = snd->toSampleBuffer();
16 //convert it back to a Samples obj.
17 var snd2 = sbuf->toSamples();
18
19 //create a new SampleBuffer by the 'new' operator.
20 var sbuf2 = new SampleBuffer(128);
```

Figure 1. *Samples* and *SampleBuffer* objects in LC.

```
01 //indexed-access to a SampleBuffer object.
02 var sb = new SampleBuffer(256);
03 for (var i = 0; i < sb.size; i+=1){
04   sb[i] = i * 2
05 }
06
07 //indexed-access to a Samples object.
08 //Samples is read-only (immutable)
09 var snd = sb->toSamples();
19 for (var i = 0; i < snd.size; i +=1){
11   println("snd[" .. i .. "]=" .. snd[i]);
12 }
```

Figure 2. Example of indexed access in LC.

In the microsound synthesis framework, two objects, *Samples* and *SampleBuffer* directly represent microsounds. While the former is immutable and the latter is mutable, the methods to convert between these two objects are provided. The *Samples* object is mainly used to manipulate and schedule microsounds. Figure 1 shows various methods to create *Samples* objects and Figure 2 displays the examples of index access to the *Samples* and *SampleBuffer* objects.

In LC's microsound synthesis framework, microsound synthesis is performed with these microsound objects together with various methods and library functions. Figure 3 and Figure 4 depict examples of microsound synthesis in LC, just as described in [3] and [4].

```
01 //create a SampleBuffer and fill it with 256 samp
02 //sine wave * 4 cycles.
03 var PI = 3.14159265359;
03 var sbuf = new SampleBuffer(1024);
04 for (var i = 0; i < sbus.size; i+=1){
05   sbuf[i] = Sin(PI * 2 * (i * 4.0 / sbuf.size);
06 }
07
08 //create a grain, apply an envelope, resample it.
09 var tmp = subf->toSamples();
10 var win = GetWindow(1024:samp, \hanning);
11
12 var grain = tmp->applyEnv(win)->resample(440);
13 grain = grain->amplify(0.25);
14
15 //perform granular synthesis for 5 sec.
16 within(5::second) {
17   while(true){
18     WriteDAC(grain);
19     now += grain.dur / 4;
20   }
21 }
```

Figure 3. Example of synchronous granular synthesis in LC [3, p.132].

```
01 //create an array to store pregenerated grains
02 var grains = new Array(100);
03
03 //generate grains with 400-500 Hz sine waves
04 var win = GetWindow(512::samp, \hanning);
05 for (var i = 0; i < grains.size; i += 1){
06   //use a unit-gen object to create a Samples obj.
07   var src = new Sin~(i + 400);
08   var tmp = src->pread(win.dur);
09   var grn = tmp->applyEnv(win);
10   grains[i] = grn;
11 }
12
13 //perform granular synthesis for 5 sec.
14 within(5::second) {
15   while(true){
16     var idx = Rand(0, grains.size – 1);
16     PanOut(grains[idx]);
19     now += grains[idx].dur / Rand(0.5, 2);
20   }
21 }
```

Figure 4. Example of granular synthesis with pregenerated grains [4].

Figure 3 is an example of simple synchronous granular synthesis [5] in LC. As shown, the *Samples* and *SampleBuffer* objects are used to represent microsounds, and manipulations can be directly applied to these objects (lines 12-13) by method calls (see [3, p.132] for more details of the code). Note that as the *Samples* object is immutable, it can be reused and rescheduled even when the same object may overlap at a certain point in time (line 17-20), without any extra care. Generally speaking, the sound object in unit-generator languages (such as *instrument* in

[5] Synchronous granular synthesis is a kind of granular synthesis, in which "sounds results from one or more streams of grains" (i.e. stream(s) of microsounds). "Within each stream, one grain follows another, with a delay period between the grains. Synchronous means that the grains follow each other at regular intervals" [1, p.93].

Csound and *synth* in SuperCollider [14]) cannot realize such self-overlapping, because each unit-generator must maintain its own internal current status that changes as sound synthesis is performed.

In contrast to synchronous granular synthesis, asynchronous granular synthesis "scatters the grains over a specified duration within regions inscribed on the time-frequency plane" [1, p.96]. Figure 4 is an example of asynchronous granular synthesis with the grains made from the sine wave of frequency between 400-500 Hz and irregular intervals. In this example, since the grains are pre-generated beforehand and only scheduling is performed in actual sound synthesis, significantly enhanced performance efficiency can be achieved. Such abstraction of the microsound synthesis framework in LC permits describing various microsound synthesis techniques more tersely and simpler versus existing unit-generator languages (see [3] and [4] for more details).

2.4 Lazy Evaluation and Digital Sound Synthesis

2.4.1 Lazy Evaluation

Lazy evaluation is a method to evaluate programs, often seen in functional programing languages. The list of the languages that have lazy evaluation in the language specification includes Haskell [15], OCaml [16], Scala [17], and others. In a *lazy language*, a program "will not evaluate any expression unless its value is demanded by some other part of computation", whereas, in a *strict language*, a program "evaluate[s] each expression as the control flow of the program reaches it[6]" [18, p.322].

While there are not many examples, certain computer music languages adopt lazy evaluation for sound synthesis. The following sections describe such languages.

2.4.2 The Fugue Computer Music Language

Fugue [19], a computer music language developed by Dannenberg et al. as an internal domain-specific language built on XLISP [20], is an early example of computer music language applying lazy evaluation in sound synthesis.

```
01 (setf Mysound (sfload "mysound")
02 (setf Demo (scale 2.0 (seq (cue Mysound)
03                            (cue Mysound))))
04 (play Demo)
```

Figure 5. A simple example of sound synthesis in the Fugue language [19].

Figure 5 portrays a simple example code in the Fugue language, as described by Dannenberg et al. in [19]. First, the variable *Mysound* is bound to a sound stored in the file "mysound" (line 01). Line 02 creates a score consisting two copies of *Mysound* scaled by 2.0 and sets it to the variable *Demo*. However, Fugue does not evaluate this score at this point and no computation is performed. In line 03, the *play* function call forces the evaluation to produce the sample output. When this computation is performed, the computed samples are memorized in *Demo*.

Nevertheless, unlike many other lazy languages, *Fugue* does not memoize the result of intermediate computations by default [21]; lazy evaluation is employed rather as a technique to eliminate unnecessary memory allocation and signal copying to form intermediate results for the improvement of the performance efficiency [21]. Fugue was significantly extended later and renamed *Nyquist*. Nyquist also performs lazy evaluation, yet approaches sound synthesis incrementally by block processing so that it can reduce the required memory space [22], while Fugue allocates the enough memory space for the entire result and computes one-at-a-time. The intermediate computed results are not memoized by default also in Nyquist [21].

2.4.3 The Chronic Computer Music Language

Chronic developed by Brandt [12], which is an internal domain-specific language (DSL) built on OCaml for non-real-time sound processing, is another noteworthy example of a computer music language that adopted lazy evaluation for sound synthesis. As already described in Section *2.2 Software Framework for Microsound Synthesis*, Chronic has the type *'ivec'*, which is a vector of infinite size. Unlike strict languages, lazy languages can handle vectors of infinite size without any difficulty, as any value in the vector is not evaluated until it is actually required.

As seen in the example of granular sound synthesis ("*Sample vec event ivec*"), Chronic utilizes this feature to express an audio stream of infinite length, without modeling it as a data-streaming object (like unit-generators). Such a framework design fosters removal of the abstraction barrier[7] to low-level sample data, by avoiding the encapsulation of it and allowing direct access, even when handling the streaming of audio and event data.

3. DESCRIPTION OF OUR WORK

3.1 Temporal Suspension of Sound Synthesis

As described in Section *2.3: The Microsound Synthesis in LC*, the LC language provides microsound objects and related manipulations for microsound synthesis. This software design assumes that microsound synthesis techniques deal with fairly short sound particles and scheduling algorithms are performed sporadically.

Yet, while this assumption is practically justifiable in performing microsound synthesis techniques, when manipulating a very large *Samples* object beyond microsound time-scale, real-time sound synthesis can be temporarily suspended, because the computation is performed eagerly

[6] This evaluation strategy of strict languages is also often referred to as *eager evaluation.*

[7] Abstraction barriers "isolate different 'levels' of the system." "At each level, the barrier separates the programs … that use the data abstraction from the programs … that implement the data abstraction" [24, p.88]

in the current version. It can consume too much time to manipulate very large *Samples* objects, and may fail to meet the real-time deadline for sound synthesis.

For example, if each DSP cycle requires 256 frames of samples for the audio output under the 44.1 kHz sample rate, this samples corresponds to about 5.8 msec. Yet, if a large *Samples* object, like one consisting of 4,410,000 (= 100 sec/44.1kHz) samples, is manipulated during one DSP cycle, it easily consumes more than 5.8 msec. This can suspend the real-time sound synthesis for a while; while this type of situation is beyond what the microsound synthesis framework in LC assumes, it is still desirable to avoid such temporal suspension. Figure 6 shows a simple example that would bring about such temporal suspension in LC.

```
01 //read three second from the buffer No.0.
02 var snd = ReadBuf(0, 1::second);
03 //play it.
04 WriteDAC(snd);
05
06 now += 0.5::second; //0.5 sec wait.
07
08 //resample it to 600 * 44100 samples.
09 //this temporarily suspend real-time synthesis,
10 //as it consumes too much CPU time.
11 var tmp = snd->resample(600 * 44100);
```

Figure 6. Example to temporarily suspend the DSP.

3.2 Lazily Manipulating Microsound Objects

We adopted lazy evaluation to solve this problem of temporal suspension. As the evaluation of a microsound object is delayed until it is actually required (e.g., for the audio output or for the access to the samples within the microsound object) and only the required part of the microsound object is computed, leaving the rest of samples uncomputed, the pause time imposed during evaluation can be significantly reduced.

We implemented a simple software framework in C++ from scratch to evaluate how effective this technique can be in practice; as it is clearly more desirable to avoid the influence from any other factors (such as memory allocation and garbage collection) when measuring the performance efficiency, we opt to not directly integrate this technique into an existing language.

Figure 7 shows the excerpt of the definition of the abstract class for microsound objects. As is present in line 06, each samples within the *Microsound* instance can be accessed by the overloaded operator '[]'. To exclude the influence of memory allocation, which can consume a significant amount of the CPU time from the measurement of the performance efficiency, in this testing framework, all the memory was allocated with the constructor method[8], and the *init* method performs any other initialization required for sound synthesis. The *init* method also calls the *_init* method of the subclass.

```
01 class Microsound {
02 public:
03   Microsound(void);
04   virtual ~Microsound(void);
05   virtual int64_t   size(void) = 0;
06   virtual lc_sample operator[](int64_t index) = 0;
07   virtual void      init(void) final;
08 };
```

Figure 7. The base class for all the microsound classes.

```
01 //the sinewave class (The eager-evaluation)
02 //called when the instance is initialized.
03 void MSEagerSineWave::_init(void)
04 {
05   double phase = 0.0;
06   double phaseInc = 2.0 * PI * freq / gSampleRate;
07
08   int64_t size = this->size();
09   for (int64_t i = 0; i < size; i++){
10     this->buf[i] = sin(phase) * amp;
11     phase += phaseInc;
12    if (phase > 2.0 * PI || phase < -2.0 * PI){
13        phase = fmod(phase, 2.0 * PI);
14     }
15   }
16   return;
17 }
```

Figure 8. Excerpt of the sinewave microsound class implementation (the eager-evaluation).

```
01 MSLazySineWave::_init(void)
02 {
03   this->pahseInc = 2.0 * PI * freq / gSampleRate;
04 }
06 lc_sample MSLazySineWave::operator[](int64_t idx)
07 {
08   int64_t bitmapIndex = index / MSBLOCK_SIZE;
09   if (this->computedBlocks[bitmapIndex]){
10     return this->samples[idx];
11   }
12
13   int64_t start = bitmapIndex * MS_BLOCK_SIZE;
14   int64_t end   = start + MS_BLOCK_SIZE;
15   if (end >= this>samples.size()){
16     end = this->samples.zie();
17   }
18
19   double phz = fmod(phaseInc * start, 2.0 * PI);
20   for (int64_t i = start; i < end; i++){
21     this->samples[i] = (lc_sample)(sin(phz)* amp);
22     phz += phaseInc;
23   }
24   this->computedBlocks[bitmapindex] = true;
25   return this->samples[idx];
26 }
```

Figure 9. Excerpt of the sinewave miscrosound class implementation (the lazy-evaluation).

Figure 8 and Figure 9 present the excerpts from the implementation of the sinewave microsound class. Each figure shows the versions that perform eager evaluation and lazy evaluation, respectively. In Figure 8, before the instance is created[9], the *_init* method is called to fill the buffer to with the waveform of the sine wave at once. The computed samples can be accessed by the '[]' operator.

On the contrary, in Figure 9, the *_init* method of the lazy evaluation version only sets up the phase increment parameter, and does not compute any samples. Yet, when the indexed access is performed, it verifies whether the sample

[8] Based on the fact that the CPU time used for memory allocation is often less predictable and can consume significant amounts of time, real-time computer music software needs to take extra care, as discussed by Dannenberg and Bencina [24].

[9] To exclude the CPU time for the memory allocation from the performance evaluation, the required memory space was allocated before real-time sound synthesis began. The other components performing signal processing were executed during real-time synthesis.

value at the index is previously evaluated. If already evaluated, the method simply returned the memoized value. If not, it computes the sample value and return.

Note that one block of samples is computed altogether by block processing to improve performance efficiency. The internal buffers within the microsound objects are divided into a number of the blocks with fixed size, and each block is computed at once, when any access to a sample value within the block is made. The flag if the block is already computed or not is maintained in the separate bitmap (as observed in lines 08-11 of Figure 9). Since the std::vector<bool> uses only one byte for eight elements of the boolean type, each byte can manage the statuses of eight blocks.

While lazy evaluation can distribute the cost of the computation among the DSP cycles by computing on demand, such block processing also contributes to increasing computational efficiency when the computation is demanded; thus, the pause time in the creation of microsound objects can be significantly reduced.

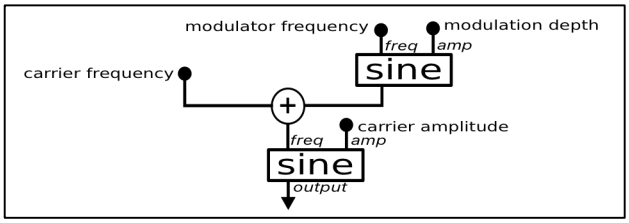


Figure 10. A simple FM synthesis instrument

```
01 //building a FM synth sound (carrier freq = 1000,
02 //modulator freq = 6, modulation depth = 15)
03 int64_t size = GetSampleRate() * 5;
04 mod   = new MSLazySineWave  (size, 6, 15);
05 cfreq = new MSLazySig       (size, 1000);
06
07 //add the carrier freq and the modulater output.
08 freq  = new MSLazyArithmetic('+', mod, cfreq);
09
10 //generating the final output samples.
11 amp   = new MSLazySig       (size, 1.0);
12 out   = new MSLazySineWave2 (freq, amp);
```

Figure 11. Performing FM synthesis with microsound objects.

MSEagerSig and MSLazySig: is a microsound object with the constant values.
MSEagerWhiteNoise and MSLazyWhiteNoise: is a microsound object filled with white noise.
MSEagerSineWave and MSLazySineWave: is a microsound object filled with a sine wave. The frequency and amplitude parameters are specified by floating point values.
MSEagerSineWave2 and MSLazySineWave2: is a microsound object filled with a sine wave. The frequency and amplitude parameters are given by other microsound objects.
MSEagerArithmetic and MSLazyAritmetic: creates a new microsound objects. By performing an arithmetic operation (+,-,*,/) to two input microsound objects and
SUEnvelope: creates a new microsound with the given envelope parameters.

Table 1. The list of the available unit-generators

Table 1 lists the microsound objects prepared for the assessment of the performance efficiency. While there are few objects, these are enough to perform basic tasks, such as additive synthesis and FM synthesis. For instance, a simple FM synthesis (as portrayed in Figure 10) can be carried out by combining these objects as seen in Figure 11.

4. PERFORMANCE MEASUREMENT

4.1 The Test Environment

As described in the previous section, the testing software framework was written exclusively in C++ from scratch, independently of any existing computer music software, so that we could exclude other factors regarding the language implementation as possible. We used the *clock* library function so that the exact CPU time spent for the sound synthesis would be measured without the influence of task-switching.

All the tests were performed on a Mac Book Air 2015 (11-inch, Intel Core i5 1.6GHz, 4GB Memory, OS X El Capitan). The code was compiled with the '-Ofast' option (the fastest aggressive optimization) with the Apple LLVM7.1 compiler. The I/O block size for the sound output was set to 256 samples, and the sample rate was set to 44.1kHz. The block size for the lazy evaluation version of microsound objects was set to 256 samples.

4.2 The Test Tasks

Table 2 outlines the test tasks for the evaluation. Each task was performed five times for both eager evaluation and lazy evaluation to measure the worst-case CPU time, the best-case CPU time, and the average CPU time. Each task generated the sounds of 10 sec, 30 sec, and 120 sec.

Task #1: White Noise with Ring Modulation A white noise sound is scaled by a sine wave sound.
Task #2: Additive Synthesis Additive synthesis consisting of four sine wave sounds and one envelope applied to the entire sound.
Task #3: FM Synthesis A simple FM synthesis sound as described in Figure 10.

Table 2. The test tasks for the performance measurement.

4.3 The Test Results

Table 3 details the results of the test tasks. All the results in the table are in milliseconds. The numbers in the '*at the initialization'* section refer to the worst-case CPU time (max), the best-case CPU time (min), and the average CPU time in the DSP cycle when the microsound objects are initialized.

The numbers in the 'rest' section are the CPU times spent in the rest of the DSP cycles until the end of the sound. The total CPU time is the entirety of the CPU time spent finishing the evaluation of the microsound objects (the average of the five trials for each task).

	At the initialization			*The rest*			*Total CPU time (ms)*
	avg (ms)	*max (ms)*	*min (ms)*	*avg (ms)*	*max (ms)*	*min (ms)*	
The Fastest-Aggressive Optimization (-Ofast)							
Task 1: White Noise With Ring Modulation							
10 sec							
Eager	21.334	25.090	17.494	0.008	0.029	0.001	34.762
Lazy	0.088	0.092	0.081	0.029	0.105	0.012	50.518
30 sec							
Eager	63.685	72.217	55.735	0.009	0.115	0.002	112.537
Lazy	0.053	0.062	0.046	0.032	0.158	0.011	164.912
120 sec							
Eager	208.183	224.036	196.261	0.007	0.053	0.001	350.470
Lazy	0.077	0.092	0.055	0.038	0.192	0.010	794.457
Task 2: Additive Synthesis							
10 sec							
Eager	61.828	80.981	53.277	0.008	0.030	0.002	75.798
Lazy	0.089	0.107	0.052	0.083	0.489	0.033	142.842
30 sec							
Eager	201.302	220.587	179.687	0.008	0.053	0.001	241.545
Lazy	0.095	0.137	0.058	0.083	0.502	0.032	427.433
120 sec							
Eager	803.788	881.327	679.922	0.014	0.164	0.002	1092.172
Lazy	0.122	0.154	0.089	0.095	0.543	0.032	1954.035
Task 3: FM Synthesis							
10 sec							
Eager	27.320	33.822	24.116	0.008	0.143	0.002	41.423
Lazy	0.070	0.094	0.048	0.033	0.116	0.014	57.505
30 sec							
Eager	84.239	106.615	67.021	0.008	0.050	0.001	124.650
Lazy	0.070	0.095	0.050	0.034	0.161	0.014	173.607
120 sec							
Eager	352.669	406.257	288.634	0.013	0.284	0.002	621.765
Lazy	0.106	0.124	0.076	0.040	0.377	0.012	817.449

Table 3. The Test Results

5. DISCUSSION

5.1 The Evaluation of the Test Results

Overall, the test results indicated what was theoretically expected. In the eager evaluation version, as the sizes of the microsound objects became larger, the time costs associated with the evaluation at the initialization increased almost proportionally. For example, the average CPU time per DSP cycle (avg) observed for the FM synthesis task by eager evaluation was 27.320 msec for the 10 sec sound, 84.239 msec for the 30 sec sound, and 352.69 sec for the 120 second. The time cost after the initialization stayed constant regardless of the duration of the sound, because it only retrieved the sample data already computed at the timing of the initialization. As shown in Table 3, it ranged from 0.08 msec to 0.014 msec.

In contrast, lazy evaluation significantly diminished the pause time imposed by the manipulation of microsound objects, even when the sizes of the microsound objects were far beyond the microsound time-scale as in these test tasks. The CPU cost in each DSP cycle remained mostly constant regardless of the size of the microsound object. For example, the CPU time spent at the initialization ranged between 0.053 msec and 0.122 msec, and the average CPU time for the rest stays almost constant for each task (0.029 – 0.33 msec for task 1, 0.083-0.095 msec for task 2, and 0.033 – 0.040 msec for task 3). While it was observed that the total CPU time costs were equal to roughly 1.5-2 times as much as the eager evaluation version, the significant reduction of the pause time achieved by lazy evaluation is quite favorable for computer music applications. To meet the real-time deadline is the most important criterion for real-time sound synthesis.

Thus, the adoption of lazy evaluation in our technique significantly reduced the pause time and therefore the temporal suspension, as described in [3] and [4], can be avoided. While our original microsound synthesis framework in the LC language assumed only the use for microsound synthesis techniques, our lazy evaluation technique can aid in enlarging the potential application domain of his microsound synthesis framework, towards more general sound synthesis techniques.

5.2 The Difference from Existing Works

As discussed in the Related Work section, there are not many examples of the previous literature utilizing lazy evaluation for digital sound synthesis, and these existing works significantly differ from that which we presented here. Both Fugue and Chronic perform non-real-time sound synthesis. Fugue (and Nyquist) adopts lazy evaluation to reduce memory allocation to improve the overall performance efficiency and Chronic employs lazy evaluation to express a data stream as an infinite-length vector; in contrast, our work adopts lazy evaluation to reduce the pause time in real-time sound synthesis, by distributing the computational cost among the DSP cycles.

5.3 The Memory Usages

Memory usage is one of the issues needing to be discussed regarding our technique. Indeed, this is one of the reasons Nyquist utilizes block processing and does not memoize the intermediate results, as the available physical memory space was not as large as is common place today and the allocation of large memory can lead to frequent paging to/from the external storage, significantly damaging the performance efficiency. However, computer systems have much larger physical memory space nowadays, and the audio data is not overly large for fitting in the physical memory space in most situations. Moreover, while our current implementation holds intermediate results, if the computer music language has the garbage collection feature, the intermediate results unreachable from the program can be automatically released[10].

5.4 The Extended Discussion

One of the possible extension for this technique is to greedily evaluate the samples before they are actually needed. The samples can be computed in other threads in parallel with the audio thread. Generally speaking, when performing real-time sound synthesis, the audio thread periodically computes the audio output with a certain interval so that it can coordinate its computation with the progress of real

[10] If garbage collection utilizes reference counting [25] (or combine it with another garbage collection mechanism), such memory space can be released immediately when it becomes unreachable. This would make the reuse of the memory space allocated for microsound objects faster.

time. Such temporal behavior is important to realize interactive control in a computer music system.

However, while the sound synthesis is performed during this periodic computation in the audio thread in the current test environment, it is also possible to evaluate the sample values left uncomputed within microsound objects in background threads in parallel. This would not cause much damage to the temporal behavior of the audio thread, since it is not necessary to synchronize these threads for the audio computation, if microsound objects are immutable (as in LC's *Samples* object). Even when the audio thread and background threads compute the same block of samples and evaluate the samples data simultaneously, the computed results are identical because of immutability; hence, the evaluation can be performed in parallel without any problem and significant improvement of the computational efficiency can be expected with multithreading.

6. CONCLUSIONS AND FUTURE WORK

In the present work, we proposed a solution to the problem of temporary suspension of real-time sound synthesis as seen in the microsound synthesis framework in the LC language. By adopting lazy evaluation to manipulations of microsound objects, the pause time imposed by the manipulations of microsounds can be significantly diminished and, thus, temporary suspension can be avoided. Such a feature is quite favorable for utilizing the software abstraction of the microsound synthesis framework in more general applications beyond microsound synthesis. When combined with the reusability of previously computed samples (e.g., reusing the pre-generated microsounds), a significant improvement can be expected in the performance efficiency in real-time sound synthesis, which existing unit-generator languages can hardly emulate.

For the future research, we are planning to extend the technique, as alluded to earlier, to evaluate samples that are still not yet computed, greedily in multithreads, for further improvement in the performance efficiency.

Acknowledgments

We gratefully thank Prof. Roger Dannenberg for providing us the detailed information on the *Fugue* language.

7. REFERENCES

[1] C. Roads, Microsound, MIT Press, 2004.
[2] C. Roads, The Computer Music Tutorial, MIT press, 1996
[3] H. Nishino, LC: A Mostly-strongly-timed Prototype-based Computer Music Programming Language that Integrates Objects and Manipulations for Microsound Synthesis, Ph.D. Thesis, National University of Singapore, 2014
[4] H. Nishino, et al., "The Microsound Synthesis Framework in the LC Computer Music Programming Language." In Computer Music Journal, 39(4), MIT Press, 2016, pp.49-79.
[5] C. Roads, "Introduction to Granular Synthesis," Computer Music Journal 12(2), MIT Press, 1988, pp.11-13
[6] X. Rodet, "Time-Domain Formant-Wave-Function Synthesis," In Spoken Language Generation and Understanding. Springer, pp.429-441.
[7] J.M. Clarke, et al. "VOCEL: New Implementations of the FOF Synthesis Method," In Proc. ICMC, pp.357-371.
[8] T. Wishart, Audible Design. Orpheuse the Pantomime, 1994.
[9] D. Gabor, "Lectures on Communication Theory," Technical Report 238, Massachusetts Institute of Technology, Research Laboratory of Electronics, 1952.
[10] E. Brandt, "Temporal Type Constructors for Computer Music Programming," In Proc. ICMC, 2000.
[11] R. Boulanger. The Csound Book: Perspectives in Software Synthesis, Sound Design, Signal Processing, and Programming, MIT Press, 2000.
[12] E. Brandt, "Temporal Type Constructors for Computer Music Programming," PhD dissertation, Carnegie Melon University, 2008
[13] A.F. Blackwell and T.R.G. Green, "Notational systems - the cognitive dimensions of notation framework ", In HCI models, theories and frameworks: Toward a multidisciplinary science, Morgan Kaufmann, 2003, pp. 103–134.
[14] S. Wilson., et al., The SuperCollider Book. MIT Press, 2011
[15] S.P. Jones, Haskell 98 Language and Libraries: the Revised Report. Cambridge University Press, 2003.
[16] Y. Minsky, et al., Real World OCaml: Functional Programming for the Masses, O'Reilly, 2013.
[17] M. Odersky, et al., An Overview of the Scala Programming Language., No. LAMP-REPORT-2004-006, École Polytechnique Fédérale de Lausanne, 2004.
[18] W. Apple and P. Jens, Modern Compiler Implementation in Java. MIT Press, 2002.
[19] R. B. Dannenberg., et al., "Fugue: A functional language for sound synthesis," In Computer 24.7, 1991 pp.36-42.
[20] R. B. Dannenberg. In an email exchange, dated Apr 18th, 2016.
[21] D. M. Betz, Xlisp: An Object-Oriented Lisp, Version 2.1., Apple, 1989
[22] R.B. Dannenberg, "The Implementation of Nyquist, A Sound Synthesis Language," In Computer Music Journal 21.3, 1997, pp.71-82.
[23] H. Abelson., and G. J. Sussman, Structure and Interpretation of Computer Programs, MIT Press, 1996.
[24] R. B. Dannenberg, and R. Bencina, "Design patterns for real-time computer music systems," ICMC 2005 Workshop on Real Time Systems Concepts for Computer Music, In Proc. ICMC, 2005.
[25] G. E. Collins, "A method for overlapping and erasure of lists," Communications of ACM, 3 (12), pp.655-657

SPECULATIVE DIGITAL SOUND SYNTHESIS

Hiroki Nishino
Imagineering Institute, Malaysia &
Chang Gung University, Taiwan
hiroki.nishino@acm.org

Adrian David Cheok
Imagineering Institute, Malaysia &
City University London, United Kingdom
adrian@imagineeringinstitute.org

ABSTRACT

In this paper, we propose a novel implementation technique, speculative digital sound synthesis, as a practical solution for the tradeoff between computational efficiency and sample-rate accurate control in sound synthesis. Our technique first optimistically assumes that there will be no change to the control parameters for sound synthesis and computes by audio vectors at the beginning of a DSP cycle. Then, after the speculation, when any change is made in the same cycle, it recomputes only the necessary amount of the output.

As changes to control parameters are normally quite sporadic in most situations, recomputation is rarely performed. Hence, the computational efficiency can be maintained mostly equivalent to the computation by audio vectors without any speculation, when no changed is made to the control parameters. Even when any change is made, the overhead can be reduced since the recomputation is only applied to those sound objects that had their control parameters updated, and the output samples in the past are not recomputed.

Thus, our speculative digital sound synthesis technique can provide both sample-rate accurate control in sound synthesis and better performance efficiency by the audio vectors in most practical situations. The tradeoff between these two issues has been a long-standing problem in computer music software design.

1. INTRODUCTION

While the development of faster CPUs in the past decades realized real-time sound synthesis even on laptop computers, the desire for better computational efficiency in digital sound synthesis still remains for many reasons. Computer music practices of our time can involve CPU-intensive sound synthesis techniques. Real-time sound processing of live instruments requires lower audio latency. Audiovisual performances that involves video processing can consume a significant amount of CPU time.

On the other hand, recently there has also been an increasing demand for sample-rate accurate timing behavior recently among computer musicians. For example, it is often required to schedule microsounds with sample-rate accurate timing for microsound synthesis techniques [1]. Inaccuracy in scheduling can lead to a different sound output from that theoretically expected, which can often be audible to human ears. For another example, as Lyon discussed, “a pulsation may feel not quite right when there are a few 20s of milliseconds of inaccuracy in the timing from beat to beat” and “smaller inaccuracy, though rhythmically acceptable, can still cause problems when sequencing sounds with sharp transients, since changes in alignment on the order of a couple of milliseconds will create different comb filtering effects, and the transients slightly realign on successive attacks” [2].

However, there has been a long-standing tradeoff between these two issues, computational efficiency in digital sound synthesis and sample-rate accuracy in timing precision, in computer music software and languages. Generally, the utilization of audio vectors (arrays of sample data) [3, p. 467] is the most popular implementation technique seen in many computer music software and languages to improve computational efficiency in digital sound synthesis. Yet, as output samples within the audio vector are computed at once, the change made to the control parameters can be reflected to the sound synthesis only at the beginning of this computation. Hence, this results in the *control rate* updating the sound synthesis parameters to be lower than the *audio rate* (sample rate) [3, p. 467].

The use of audio vectors, at the cost of sample-rate accurate timing behavior, has been generally considered acceptable in the past, because, given a sufficiently high control rate, human ears are not so sensitive to the slight details in sound output, even when the control rate is set lower than the audio rate. Yet, some of recent computer music programming languages performs sound synthesis using sample-by-sample computation to achieve sample-rate accuracy in timing behavior at the cost of computational efficiency, since there is a significant demand for sample-rate accuracy to support recent computer music practices, as described earlier.

As Moore’s law may end [4], it may not be expected that CPUs are significantly faster in the near future. Therefore, it is of significant importance to develop new implementation techniques to achieve computational efficiency even when sample-rate accuracy in timing precision is required. In this paper, we describe a new technique we developed that adapts *speculative computation* [5] to digital sounds synthesis. The technique speculatively

computes the audio output using the audio vectors, assuming there is no update to sound synthesis parameters in a DSP cycle, and recomputes only the necessary number of samples for those sound objects that had their sound synthesis parameters updated in the same cycle.

Since, in most practical situations, changes to sound synthesis parameters are made sporadically, and do not occur many times in one DSP cycle, our speculative computation technique can achieve computational efficiency equivalent to the existing techniques to utilize audio vectors when no change is made in a DSP cycle. It also can reduce the overhead for the recomputation, since the recomputation is applied only to the sound objects with updated parameters and the past samples are not recomputed. In other words, our technique provides a practical solution for the problem of the traditional tradeoff between computationally efficiency and sample-rate accurate timing behaivour.

2. RELATED WORK

2.1 Audio Vectors

The simplest approach to implement digital sound synthesis modules is to write a function that computes only a single sample at a time. Figure 1 describes the code example in the C programming language of a table-lookup oscillator that takes this approach, as described by Lazzari in [3, p. 466].

```
A table-lookup oscillator function (1)
01:float oscil(float  amp,
02:            float  freq,
03:            float* table,
04:            float* index,
05:            int    len,
06:            float  sr)
07:{
08:  float out;
09:  out = amp * table[(int) *index];
10:  *index += freq * len / sr;
11:  while(index >= len) index -= len;
12:  while(index < 0) index += len;
13:  return out;
14:}

A processing loop to generate a signal
01:for (int i = 0; i < durs; i++){
02:  out[i] = oscil(0.5f, 440.f, wtab, &ndx);
03:}
```

Figure 1. A table-lookup oscillator function that generates a single sample output at each call[1] [3, p. 466].

While the Figure 1 example is quite simple in its implementation, the code, however, is not very computationally efficient, due to the overhead imposed by the function call. Generally, when a function call is made, a computer program must prepare a new *stack frame* (or *activation record*) for a return address, local variables, parameters, and other temporaries [6, Chapter 7], and this imposes the additional overhead that damages computational efficiency in digital sound synthesis. The CPU cache miss may also occur when jumping to the memory space where the code of the function resides. Some may consider that such an overhead can be avoided by using the *macro substitution* [7, p. 89] or *inline function expansion* [8, p. 310]. Yet, such optimization techniques can be applied only at compile time; therefore, it is not applicable to computer music languages, which are normally required to build sound objects (e.g., *instrument* in CSound [9] or *synth* in SuperCollider [10]) dynamically at runtime.

To avoid this overhead by the function call, many computer music programs process audio in blocks by vectors of samples (audio vector). Figure 2 describes an example of this implementation technique as described by Lazzzari in [3, p. 467]. As shown, the processing loop is implemented inside the table-lookup oscillator function to process the output samples at once within the function so that the overhead by the function call can be avoided.

In practice, the control parameters for sound synthesis (such as *amp*, *freq*, *table,* and *index* in Figures 1 and 2) and the DSP function often compose a complex data type (e.g., *structure* or *class* in C++) and provides a more general interface (or a type signature of a function/method shared by other sound synthesis modules) so that sound synthesis graphs can be easily constructed at runtime, regardless of the actual type of sound synthesis modules.

```
A table-lookup oscillator function (2)
01:float oscil(float* output,
02:            float  amp,
03:            float  freq,
04:            float* table,
05:            float* index,
06:            int    len,
07:            int    vecsize,
08:            long   sr)
09:{
10:  // increment
11:  float incr = freq * length / sr;
12:
13:  // processing loop.
14:  for (int i = 0; i < vecsize; i++){
15:      // truncated lookup
16:      output[i] = amp * table[(int)(*index)];
17:      *index += incr;
18:      while(index >= len) index -= len;
19:      while(index < 0) index += len;
20:  }
21:  return *output;
22:}
```

Figure 2. A table-lookup oscillator function that utilizes the audio vectors[2] [3, p. 466].

Indeed, the utilization of audio vectors in digital sound synthesis is quite traditional in computer music; *Music V,* developed in 1966 by Mathews et al. [11], is known to be the first language that introduced such "block processing of data among its many extensions" [12]. While the use of audio vectors can significantly improve the computational efficiency, however, there is a side effect that causes a lower control rate than sample rate, as will be discussed in the following subsections.

2.2 Audio Rate and Control Rate

By observing the use of the Music 360 language [13], Vercoe found that "up to 50% of music signal processing is aimed at shaping loudness and pitch contours, – func-

[1] The *oscil* function has a function prototype with default arguments: *float oscil(float amp, float freq, float* table, float* index, int len-1024, float sr=44100);* [3, p. 466].

[2] Assuming the constants in the code (*def-len,def_vecsize, and def_sr)* are previously defined.

tion essentially of acoustic control that need not be controlled at audio rates" [14]. He developed *Music 11* in 1982, which is known to be the first computer music language in history that introduced the distinction between *audio signal* and *control signal*. As Vercoe discussed, such 'acoustic control' parameters (e.g., envelope shaping, pitch control, vibrate, etc.) can be updated with less expensive data rates without damaging perceptual quality to human ears. Computing such parameters in a lower data rate than sample rate can contribute to significant improvement in computational efficiency. These two different data rates are normally referred to as *audio rate* and *control rate.*

Indeed, the implementation examples in Figures 1 and Figure 2 already illustrate the concept of 'control rate.' In the Figure 1 example, as the oscillator function computers only a single sample at each call, the *freq* and *amp* parameters can be updated when computing each sample. In contrast, in the Figure 2 example, the oscillator function can update these control parameters only before the processing loop computes the output samples. Hence, in the Figure 2 example, the *freq* and *amp* parameters may be seen as control-rate parameters, as they are updated in a lower rate than the audio rate.

Even if we modify the *oscil* function so that it can receive the audio vectors for these two parameters, these audio vectors must be prepared as arguments before each function call. This means these control parameters cannot be updated while the function is processing the audio vectors. Thus, normally, the use of the audio vectors leads to the existence of a control rate that is lower than the audio rate.

2.3 The Issues Regarding the Use of Audio Vectors in Digital Sound Synthesis

In this section, we discuss several issues related to the utilization of audio vectors in computer music systems, to clarify the problem domain that our novel technique of speculation digital sound synthesis handles.

2.3.1 Audio Vectors and Control Rate

As described earlier, while Music 360 already introduced the audio vectors, it still computed all the control parameters for sound synthesis also by the audio vectors. In contrast, Music 11 improved the computational efficiency by introducing *control signals,* in which such 'control-rate' parameters are implemented as scalar values, not as audio vectors[3].

As suggested by this difference between the implementation of Music 360 and Music 11, the existence of the control rate and the use of audio vectors are two separate issues, although there is a strong association between them. For instance, assume a computer music system first updates to all the sound synthesis parameters before processing the sound output, and then performs sample-by-sample computation in the main processing loop until it produces enough number of samples for the current DSP cycle. Such a computer music program has a control rate but does not utilize audio vectors at all. Thus, the association between the use of audio vectors and the existence of the control rate is not an intrinsic issue but is caused by how the digital sound synthesis is implemented.

As will be described in the later sections, our speculative digital sound synthesis technique utilizes the audio vectors but still provides sample-rate accuracy in timing behavior. Regardless of the involvement of audio vectors, the control rate can be equal to the audio rate.

2.3.2 Minimum Feedback Time

Another issue regarding the use of audio vectors is that it imposes a limitation on the minimum feedback time. A sound object composed of sound synthesis modules cannot perform the feedback that is shorter than the size of the audio vectors[4]. Note the issue of minimum feedback time is not directly associated with the control rate. Generally, the control rate is about the minimum interval that can make a valid change to control parameters. This update of the control parameters may be made by control signals (emitted by the control-rate unit-generator) or by the user code external to the sound synthesis graph.[5] In contrast, the limitation on the feedback time is about data-streaming of sample values within a sound object, and there is a significant difference between these two issues, while both can be caused by the use of audio vectors.

Our speculative digital sound synthesis techniques aim at the improvement of the control rate without significant damage to computational efficiency, but does not intend to remove the limitation on the minimum feedback time.

2.4 Speculative Computation

Speculative computation is "an implementation technique that aims at speeding up the execution of programs, by computing pieces of code in advance, possibly in parallel with the rest of the program, without being sure that these computations are actually needed" [6]. The idea of speculative computation (or speculative execution) can be seen in various fields in computer science. For instance, while it is not aiming for the improvement in the performance efficiency, a *backtracking parser*[6] [15], which is a parsing technique invented around 1970, speculatively performs parsing of the input text, and "if the first attempt failed, it rewinds the input and then attempts the next

[3] According to Miller Puckette, in an email exchange dated Apr 6th 2016.

[4] For instance, assume that the size of the audio vectors is two. When computing the first sample in the output audio vector, the next sample to compute with the feedback is also in the same vector, and must be computed before sending the first sample to the feedback loop. Thus, the minimum size of the feedback is two in this case.

[5] One may argue a synthesis patch, as seen in PureData [19], integrates sound synthesis algorithms and control algorithms into one sound object (i.e., 'a patch'). In this case, simply interpret 'sound object' as 'sound synthesis graphs composed of DSP objects'.

[6] A *parser* is a computer program that performs syntactic analysis of the phrase structure of the input program text. It often translates the input program text to abstract syntax trees. The trees are used in the latter phases of compilation.

alternative" [16, p.68]. Such parsing techniques as *memoizing parsing* [17] and *packrat parsing* [18] belong to the same category, but with significant improvement in the performance efficiency in comparison to the original backtrack parsing.

Memory and files are also often predicted and prefetched in many systems to avoid the latency [20][21]. Furthermore, the recent CPUs often predict the execution path and performs speculative execution of code to improve the computational efficiency [22]. Speculation is also applied for parallelization. Thread-level speculation (TLS) "enables the compiler to optimistically create parallel thread despite uncertainty as to whether those threads are actually independent"[7] [23]. Speculative synchronization applies the same concept to barriers, locks, flags, etc. for thread synchronization [24]. Thus, while it may not be directly visible to users, the modern computer system adopts the concept of speculative execution in many important components, mostly to improve the performance efficiency.

3. DESCRIPTION OF OUR TECHNIQUE

3.1 Overview of Our Technique

Figure 3 illustrates the overall algorithm of our technique to speculatively perform digital sound synthesis. We assume that the computer music system using this technique is based on logical time with sample-rate accuracy, which corresponds to the number of the samples processed since the computer music system began its execution. The system repeatedly executes this algorithm at every DSP cycle.

As shown, this technique first processes all the tasks scheduled right at the current logical time (in the beginning of the DSP cycle). Then, it computes the output samples using the audio vectors with the size required for this DSP cycle at once, optimistically assuming there will be no update to the control parameters in this DSP cycle; thus, the performance efficiency at this phase can be the same as normal digital sound synthesis with audio vectors, as used in many computer music programs.

After this speculative execution phase, the algorithm performs other tasks scheduled in the same DSP cycle one-by-one in ascending order sorted by logical time. If any change has been made to a control parameter that affects the output result in this stage, before advancing the logical time to process the other tasks scheduled in the future (in logical time), the recomputation is performed for those sound synthesis objects with updated parameters. Similarly, in the first phase, it is optimistically assumed that there is no further update to the control parameters in the same DSP cycle, and that the audio vectors are recomputed to the end from the index associated with the current logical time.

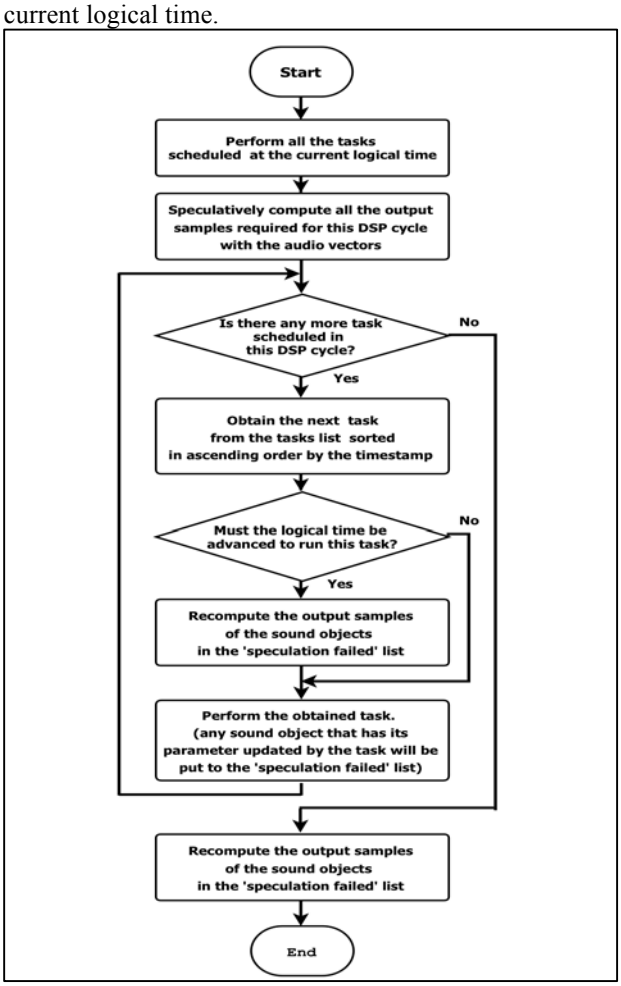


Figure 3. The overview of our algorithm to speculatively perform digital sound synthesis.

When all the scheduled tasks are performed in the previous phase, if there still remain any changes made to the control parameters that are not reflected to the output, it recomputes the necessary number of samples, but again, only for those sound objects with updated parameters. The past samples are not recomputed as they are not affected by the change.

Thus, in this algorithm, the tasks are computed with sample-rate accuracy in logical time, and the changes made by these tasks can be reflected to the output with the same timing precision. Hence, while it utilizes the audio vectors, the behavior of the system can achieve sample-rate accurate timing precision. (but note that there is still a restriction on the feedback time due to the involvement of the audio vectors).

Such adaption of speculative execution in digital sound synthesis allows equivalent performance efficiency to normal digital sound synthesis with the audio vectors when the speculation does not fail. It can also achieve less damage to the performance in comparison to sample-by-sample computation in most practical situations, since

[7] For instance, as described by Stefan et al. describes [23], TLS may optimistically parallelize even a 'while' loop that cannot be statically parallelized by the compiler optimization due to the possible data dependences; These threads speculatively execute the loop in parallel, except at least one *safe thread,* which executes code without speculation. If any dependence violation occurs between threads, the threads will redo the computation, except the safe thread(s). Note that the forward progress of execution is guaranteed since there is at least one safe thread.

only those synthesis objects with updated parameters are recomputed, and the past samples, which are not affected by updating, are not recomputed. Normally, it can be expected that most computer music programs just sporadically update sound synthesis parameters and do not update control parameters of many sound objects simultaneously. In the later sections, we discuss a certain situation that may impose a large penalty when speculation fails, and also propose possible solutions.

3.2 Implementation

We implemented our speculative digital sound synthesis technique to investigate whether this technique can provide both computational efficiency and sample-rate accurate timing precision at once, as it is expected in practical situations. Considering the popularity of the unit-generator concept among computer music languages, the sound synthesis framework is designed with unit-generators, while it provides only several simple unit-generators required for the experiments.

To measure the effectiveness of our technique, it is desirable to avoid any influence unrelated to the computational efficiency in digital sound synthesis, such as the overhead imposed by garbage collections, memory allocation, and the control tasks. For this reason, we implemented a simple sound synthesis framework from scratch in C++, a programming language without the automatic memory management feature so that we can measure only the computational efficiency in digital sound synthesis.

```
The SpecUGen class
01:class SpecUGen
02:{
03:public:
04:  SpecUGen(SpecPatch* patch, int64_t avecSize);
05:  virtual ~SampUGen(void)
06:
07:  virtual lc_sample* compute()=0;
08:  virtual lc_sample* recompute(int64_t offset)=0;
09:  virtual lc_sample* getOutput()=0;
10:
11:  virtual int64_t getAvecSize();
12:
13:protected:
14:  SpecPatch* patch;
15:  int64_t    avesSize;
16:};
```

Figure 4. The declaration of the *SpecUGen* abstract class.

```
The SpecPatch class
01:class SpecPatch
02:{
03:public:
04:  SpecPatch(LCAudioEngine* ae, int64_t avecSize);
05:  virtual ~SpecPatch(void)
06:
07:  virtual void compute();
08:  virtual void recompute(int64_t offset);
09:  virtual void setOutputUGen(SpecUGen* ugen);
10:
11:  virtual lc_sample* getOutput (void);
12:  virtual int64_t    getAvcSize(void);
13:
14: protected:
15:  int64_t        avecSize;
16:  SpecUGen*      outputUGen;
17:  LCAudioEngine* audioEngine;
18:};
```

Figure 5. The declaration of the *SpecPatch* class.

Figures 4 and 5 show the declaration of the base abstract class for all unit-generators (*SpecUGen*) and the class for sound objects (*SpecPatch*), respectively. The *SpecPatch* object is a sound object that is responsible for providing the output samples to the sound synthesis engine in each DSP cycle. As the software framework for this experiment is designed just to measure the actual performance efficiency of our technique, the design is quite simple and only pulls the output samples from the unit-generator object given by the *setOutputUGen* method call.

```
01:lc_sample* SCSineOsc::compute(void)
02:{
03:  //adjust phase for the better precision.
04:  phase = fmod(phase, 2 * LC_PI);
05:
06:  //perform speculative computation.
07:  for (int64_t i = 0; i < this->avecSize; i++){
08:    output[i]    = (lc_sample)sin(phase) * amp;
09:    pastPhase[i] = phase;
10:    phase += phaseInc;
11:  }
12:  return output;
13:}
14:
15:lc_sample* SCSineOsc::recompute(int64_t offset)
16:{
17:  //adjust phase for the better precision.
18:  phase = fmod(pastPhase[offset], 2 * LC_PI);
19:
20:  //perform recomputation.
21:  for (int64_t i = offset; i < avecSize; i++){
22:    output[i] = (lc_sample)sin(phase) * amp;
23:    pastPhase[i] = phase;
24:    phase += phaseInc;
25:  }
26:  return output;
27:}
28:
29:void SCSineOsc::setFreq(LCAudioEngine* engine,
30:                         double         freq  )
31:{
32:  //compute the new phase increment.
33:  phaseInc = 2 * LC_PI * freq / GetSampleRate();
34:  this->freq = freq;
35:  engine->notifyUpdate(this->patch);
36:
37:  return;
38:}
```

Figure 6. The implementation of the *compute, recompute, and setFreq* methods in the *SUSineOsc* class.

The *compute* method declared in Figure 5 (line 07) is used for speculative computation, which is called right after all the tasks scheduled at the beginning of the DSP cycle were executed, and the *recomputed* method (line 08) is called to recompute the output when any change is made to the control parameters of the unit-generators in the sound synthesis graph of this patch object, receiving an offset in the samples from the beginning of the current DSP cycle (the *offset* argument). The methods in the *SpecUGen* class are designed quite similarly. The *compute* method in Figure 4 (line 07) is called during the speculative computation in the beginning of DSP cycle, traversing the sound synthesis graph. The *recomputed* method is called during the recomputation.

Figure 6 illustrates the actual implementation of these methods in the *SUSineOsc2* class. As shown, it stores the phase information in the internal buffer (*pastPhase*) during the DSP cycle so that it can recover the related parameters at the point where the recomputation must begin. Note that some unit generators may not have to store the past parameters. For example, since the white noise unit-generator (*SUWhieNoise*) does not depend its output on any previous sample at all, it is not necessary to store the

past information. It should also be noted that the update of the control parameter is reported to the sound synthesis engine in the *setFreq* method (line 35) so that the patch can be added to the 'speculation failed' list.

As for the creation of a new sound object (*SpecPatch)* during the DSP cycle (not at the beginning of the cycle), it is normally required just to put the new sound object to the 'speculation failed' list. By initializing the internal buffers with zero and computing from the given offset, the *recompute* method can be called to produce the output at the timing of its instantiation and the output before the instantiaion can be zero-cleared. It should be noted that it is often even unnecessary to let the unit-generator objects handle such instantiation during the DSP cycle. For example, in the Figure 6 example, the *recompute* method of the *SCSineOsc* class does not require any special treatment, and the computation of the sine wave output can start from the given offset without any problem. Even if there is any special care for the instantiation during the DSP cycle, only one flag variable is required. If the *recompute* method is called before the *compute* method, the sound object was instantiated after the speculative computation was already performed, which means it was instantiated during the DSP cycle.

4. PERFORMANCE MEASUREMENT

4.1 The Test Environment

Since the software framework is designed just for the measurement of the actual performance efficiency of our speculative digital sound synthesis technique, it provides only several simple unit-generators, as shown in Table 1. However, these are enough to perform the test tasks described in Table 2. To compare with the existing technique, we also implemented the equivalent versions to perform sound synthesis sample-by-sample within the same framework and the normal audio-vector-based sound synthesis without speculation.

SUSig: converts a float value to an audio signal.
SUWhiteNoise: generates a white noise signal.
SUSineOsc: generates a sine wave signal. The frequency and amplitude parameters are specified by floating point values.
SUSineOsc2: generates a sine wave signal. The frequency and amplitude parameters are given as input audio signals.
SUArithmeticOp: performs the arithmetic operations (+, -, *, /) to two input signals.
SUEnvelope: generates an envelope signal

Table 1. The list of the available unit-generators

We performed experiments to compare the performance efficiency of our speculative digital sound synthesis technique, sample-by-sample sound synthesis, and popular audio-vector-based sound synthesis without speculation. We excluded other factors as much as possible, such as memory allocation, and execution of scheduled tasks for sound synthesis control, and measured only the CPU time consumed by the part of the code where digital sound synthesis is performed. All the tests were performed on a Mac Book Air 2015 (11-inch, Intel Core i5 1.6GHz, 4GB memory, OS X El Capitan). The code was complied with the '*-Ofast*' option (the fastest-aggressive optimization) with the Apple LLVM7.1 compiler for all the test tasks.

4.2 The Test Tasks

Table 2 shows the test tasks prepared for the performance measurement. These test tasks are designed under the assumption that the update to control parameters from the program external to the sound object is sporadic, as it is in most practical situations. Each task has two sub tasks, and is performed for three different implementations: (a) our speculative digital sound synthesis technique, (b) normally block processing by audio vectors without speculation, and (c) sample-by-sample processing. We used C++'s *std::mt19937*, a *Mersenne twister* random value generator class, with the same seed value. Hence, the events can be generated with the same timestamps and the same parameters for all these implementations for (a), (b) and (c).

Task #1: Sine Wave Oscillators
Ten sine wave oscillators are created.
Task #2: Additive Synthesis
Ten additive synthesis instruments are created. Each of them consists of four sine wave oscillators and one envelope applied to the entire output.
Task #3: FM Synthesis
Ten simple FM synthesis instruments are created. The unit-generator graph of this FM synthesis instrument is shown in Figure 7.
****Each of the above tasks has these two subtasks***
Sub task #1: Each instrument will update the base frequency (Task #1-#2) or the carrier and modular frequency (Task#3) to a random frequency (or random frequencies) with the interval of 50 msec, one-by-one in turn (only one instrument updates its frequency at the same time).
Sub task #2: One of the instruments is randomly picked up and will update the base frequency (Task #1-#2) or the carrier and modular frequency (Task#3), with the random intervals between 10 msec to 100 msec.

Table 2. The test tasks for the performance measurement

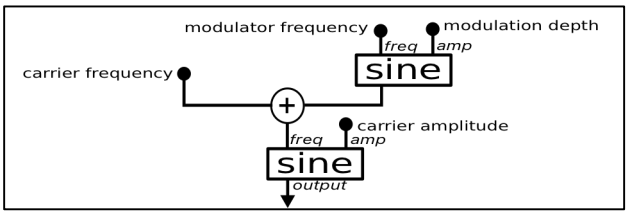


Figure 7. A simple FM synthesis instrument

To measure the exact CPU time without any influence of the task switching to other processes by the operating system, we used the *clock* C library function, which can measure only the CPU time used by the DSP process. The CPU time is measured only for the code related to the digital sound synthesis, excluding other parts, such as initialization, memory allocation, and the scheduled control tasks. Both the size of audio vectors and the DAC I/O block size are set to 256 samples. Each task continues for ten seconds and is repeated five times, to obtain the CPU time used for both the worst case and the best case (or the

max and min CPU time) among all the DSP cycles, and the average of the CPU time over all the DSP cycles.

4.3 Test Results

Table 3 shows the test results for the three tasks. The column '*recom. overhead*' shows the overhead imposed when the speculation fails (*'the CPU time for recomputation'* / '*the overall CPU time in each DSP cycle'*).

Algorithm	*avg. (msec)*	*max (msec)*	*min (msec)*	*recom. overhead*	
				avg (%)	*max(%)*
Task 1: Sine Wave Oscillators					
Sub Task 1					
sample-by-sample	0.250	1.058	0.125		
audio vector	0.128	0.575	0.061		
speculative	0.127	0.512	0.060	4.5%	22.2%
Sub Task 2					
sample-by-sample	0.224	0.719	0.126		
audio vector	0.115	0.825	0.064		
speculative	0.124	0.490	0.061	4.2%	13.9%
Task 2: Additive Synthesis					
Sub Task 1					
sample-by-sample	0.679	1.853	0.355		
audio vector	0.357	1.293	0.175		
speculative	0.345	1.174	0.177	5.0%	9.3%
Sub Task 2					
sample-by-sample	0.636	2.561	0.354		
audio vector	0.339	1.291	0.179		
speculative	0.328	1.046	0.180	4.6%	11.6%
Task 3: FM Synthesis					
Sub Task 1					
sample-by-sample	0.392	1.115	0.180		
audio vector	0.200	0.754	0.084		
speculative	0.200	0.654	0.080	5.3%	22.1%
Sub Task 2					
sample-by-sample	0.258	0.711	0.182		
audio vector	0.194	0.753	0.082		
speculative	0.189	0.798	0.084	5.7%	19.6%

Table 3. The test results.

5. DISCUSSION

5.1 The Evaluation of the Test Results

As shown in Table 3, the use of audio vectors significantly improves the performance efficiency (about twice as fast as the sample-by-sample computation). Yet, surprisingly, no significant decrease in the performance was observed in speculative implementation in comparison to the traditional audio vector implementation, while allowing sample-rate-accurate updates of the control parameters. While the overhead imposed by the recomputation is surely observed, all of the worst case (max) time, the best case time (min), and the average CPU time are almost equivalent and the difference is mostly negligible (sometimes a bit faster but still negligible) for all the test tasks. The recomputation overhead is about 5% on average and about 20% in the worst-case in all these tasks; yet, this overhead seems to not cause any damage to the overall performance efficiency in the test tasks.

Such a result is quite favorable. Our technique is likely to achieve a performance efficiency almost equivalent to the traditional audio vector implementation, since what matters for the performance efficiency in real-time sound synthesis is the worst-case execution time (to meet the sound output deadline) and the average CPU time (to give more CPU time to other processes to perform their tasks). One reason for this equivalent performance efficiency in the average CPU time may be due to our assumption that the updates to the control parameters are fairly sporadic and that the test tasks are designed upon this assumption. If the recomputation is not frequently performed, it may not affect the overall average CPU time much.

While there is certainly the necessity for a more detailed investigation, another plausible hypothesis behind the equivalent performance in the worst-case CPU time would be that the time cost imposed by the CPU cache misses may have overshadowed the cost of recomputation. While we measured only the CPU time used by the sound synthesis program using the *clock* library function, as the utilization of the cache memory by other processes can lead to unpredictable cache misses during sound synthesis, the time costs of which are imposed to the sound synthesis program, and can be quite large in comparison to the time costs for a simple DSP algorithm. This can occur anytime also in a practical situation during a live computer-music performance.

5.2 Reducing the Penalty of Speculation Failure

Our technique is based on the assumption that the update to the control parameters is fairly sporadic. Hence, if there are too many updates performed in one DSP cycle, the overhead caused by the recomputation can be unneglectable. One way to reduce this overhead is to recompute only the part of the sound synthesis graph, only where the update actually influences the output. This can be done by *data dependence analysis* [25] in the sound synthesis graph.

Another way to reduce this overhead is to divide a DSP cycle into several blocks. For instance, if the outputs of 256 samples are required for one DSP cycle, instead of making the size of audio vectors to 256, one can make it 32 and performs the speculative computation eight times. In this case, the number of the samples to recompute when speculation fails will reduce to 31 even in the worst case (about 12% of 255, when the whole output is recomputed when the audio vector size is set to 256), if there is no further update in the performed in the same DSP cycle. Further reduction of the overhead may be achieved by prediction. Generally, speculative execution often utilizes prediction to improve the performance efficiency [26]. By adaptably changing the number of samples to speculatively compute each sound object based on the past speculation results, the penalty caused by the speculation failure can be reduced.

5.3 Realizing Single Sample Feedback

As discussed earlier, the issues of control rate and minimum feedback time deal with different problem domains and are out of the scope of this paper. However, we suggest that just-in-time compilation [27] would be benefi-

cial to solve this problem by compiling the while unit-generator graph into the native code.

5.4 Multithreading

Our technique can be extended to compute the future output (e.g., for the next two seconds), and this extension can be performed by other threads independently from the main audio computation thread. When the speculation fails, the recomputation is performed first only for the current DSP cycle by the audio thread and then other threads can continue speculation. Such extension can contribute to the further improvement of the performance efficiency, especially when used with multicore CPUs.

6. CONCLUSIONS AND FUTURE WORK

We described our novel technique that speculatively performs digital sound synthesis with audio vectors and recomputes only when any update to the control parameters is made. This technique contributes to achieving both the performance efficiency and sample-rate accurate control at once in most practical situations. As described in the discussion, we observed almost no damage to our simple test tasks.

Since this tradeoff between the performance efficiency and sample-rate accurate control of sound synthesis parameters has been one of the long-standing problems in computer music software, such a speculative digital synthesis technique can be quite beneficial for next-generation computer music systems to provide both high performance real-time sound synthesis and sample-rate accurate timing behavior at once, since a significant improvement of the clock speed of CPUs can be hardly expected in the near future.

For future work, we are planning to investigate the performance efficiency in more complicated sound synthesis techniques. We also plan further extension of our speculative digital sound synthesis technique, as we discussed in the *Discussion* section, together with the implementation of this technique into the LC computer music language that we are currently developing [28].

Acknowledgments
We would like to express our gratitude to Prof. Miller Puckette for the information on the implementations of Music 360 and Music 11.

7. REFERENCES

[1] C. Roads, *Microsound*, MIT Press, 2004.
[2] E. Lyon, "A Sample Accurate Triggering System for Pd and MaxMSP," *Proc. ICMC,* 2006.
[3] R. Boulanger et al., *The Audio Programming Book*, MIT Press, 2010.
[4] M. Waldrop, "The Chips are down for Moore's law," *Nature News,* 2016.
[5] W. Apple and P. Jens, Modern Compiler Implementation in Java. MIT Press, 2002.
[6] G. Boudl, and G. Petri, "A theory of speculative computation," *Programming Languages and Systems.* Springer Berin Heidelberg, 2010, pp. 165-184.
[7] B.W. Kernighan and D.M. Ritchie, The C Programming Language (2nd edition), Prentice Hall, 1988.
[8] B. Stroustrup, The C++ programing language (4th edition), Addison-Wesley Professional, 2013.
[9] R. Boulanger et al., *The Csound book,* MIT Press, 2000.
[10] S. Wilson et al., *The Supercollider Book*, The MIT Press, 2011.
[11] M. V.Mathews, et al., *The technology of computer music*, MIT Press, 1969.
[12] B. Vercoe, "New Dimensions in Computer Music," *Trends & Perspective in Signal Processing, 2(2),* 1982.
[13] B. Vercoe, "The MUSIC 360 language for digital sound synthesis." *Proc. of the American Society of University Composers",* 6, 1971.
[14] B. Vercoe, "Computer systems and languages for audio research," *Proc. Audio Engineering Society Conference*, Audio Engineering Society, 1982.
[15] A. Birman and J. D., Ullman, "Parsing algorithm with backtrack," *Information and Control, 23(1),* 1973.
[16] T. Parr, Language Implementation Patterns, Pragmatic Bookshelf, 2009.
[17] P. Norvig, "Techniques for automatic memoization with applications to context-free parsing," *Comput. Linguist*., *17(1),* 1991.
[18] B. Ford, "Packrat parsing:: simple, powerful, lazy, linear time, functional pearl," *Proc. of ACM SIGPLAN*, 2002.
[19] M. Puckette, "Pure Data: another integrated computer music environment." *Proc. ICMC,* 1996.
[20] J. Griffioen and R. Appleton, "Reducing file system latency using a predictive approach," *Proc. USENIX summer 1994 technical conference,* 1994.
[21] H. Zhigang et al., "Timekeeping in the memory system: Predicting and optimizing memory behavior", *Proc. Computer Architecture,* 2002.
[22] A. S. Tanenbaum and T. Austion, "4.5.4 Speculative Execution", *Structured Computer Organization (The Sixth Edition)*. Pearson, 2012.
[23] J. G. Steffan, et al., "A scalable approach to thread-level speculation,", *Proc. ISCA*, 2000.
[24] J.F. Martinez and J. Torrelas. "Speculative synchronization: applying thread-level speculation to explicitly parallel applications," *ACM SIGOPS Operating System Review*, *36(5)*, 2002.
[25] D. E. Maydan et al., "Efficient and exact data dependence analysis," *ACM SIGPLAN Notices, 26(6),* 1991
[26] A. Mendelson and F. Gabbay, "Speculative execution based on value prediction," *Technical report, Technion-Israle Institute of Technology,* 1997.
[27] J. Aycock. "A brief history of just-in-time." ACM Computing Surveys, Vol.35(2), 2003.
[28] H, Nisino et al. "LC: A New Computer Music Programming Language with Three Core Features", *Proc. ICMC-SMC 2014*

A HYBRID FILTER–WAVETABLE OSCILLATOR TECHNIQUE FOR FORMANT-WAVE-FUNCTION SYNTHESIS

Michael Jørgen Olsen, Julius O. Smith III and Jonathan S. Abel
Center for Computer Research in Music and Acoustics (CCRMA), Stanford University
660 Lomita Drive, Stanford, CA 94305, USA
[mjolsen|jos|abel]@ccrma.stanford.edu

ABSTRACT

In this paper a hybrid filter–wavetable oscillator implementation of Formant-Wave-Function (FOF) synthesis is presented where each FOF is generated using a second-order filter and wavetable oscillator pair. Similar to the original time-domain FOF implementation, this method allows for separate control of the bandwidth and skirtwidth of the formant region generated in the frequency domain by the FOF synthesis. Software considerations are also taken into account which improve the performance and flexibility of the synthesis technique.

1. INTRODUCTION

Formant-Wave-Function (FOF) synthesis is a vocal synthesis technique inspired by the source–filter model of vocal synthesis that models the excitation of resonant frequencies in the vocal tract by the glottis [1, 2]. FOF synthesis has been used to create very realistic vocal sounds. This is due to the flexibility of the method in allowing the composer to shape the frequency spectrum of the sound and to morph between different vowel sounds or from vocal sounds to non-vocal sounds.

The majority of the previous implementations of FOF synthesis have focused on implementing a time-domain representation of the FOF bursts. This can be done quite cheaply, in computational terms, since table lookup can be used. However, an overlap-add scheme is needed to combine the FOF bursts into a single audio stream as the generation of a single FOF burst needs to be done independently of real-time changes of the input parameters. Also, since each burst decays exponentially, a suitable cut-off point must be chosen for when to end each particular FOF burst.

In this paper, a FOF synthesis algorithm is presented that uses a second-order filter in combination with a sinusoidal wavetable oscillator to generate the FOF bursts. The filters are triggered using an impulse train. By using a sample-and-hold mechanism in conjunction with a cycling bank of identical filters, artifact-free real-time control of input parameters can be facilitated.

The paper is organized as follows: in Section 2 prior work on FOF synthesis is reviewed; in Section 3 the second-order FOF envelope filter structure is derived; in Section 4 the details of the software implementation of the proposed algorithm are presented; in Section 5, the results are summarized and potential areas for future research are motivated.

2. PREVIOUS WORK

2.1 Speech and Vocal Synthesis

The desire to recreate human speech and singing has long fascinated humankind. After the advent of the telephone in the late 19th century, the need to send and receive the sound of human speech over transmission lines led to much research in the analysis/synthesis of human speech. Much of the original research along those lines was completed at Bell Labs during the early 20th century, yielding in particular Homer Dudley's *vocoder* ("voice coder")—a voice analysis and resynthesis device, and the Voder—a manually driven speech synthesizer demonstrated at the 1939 World's Fair [3]. The first computer-generated vocal synthesis was created with a software version of the vocoder in the 1960s, also at Bell Labs, and led to the generation of the first digital singing synthesis [4]. Later on, resonant band-pass filters were used to generate vocal synthesis [5, 6] but the technique was prone to audible artifacts being present due to the sharp discontinuity at the start of the exponential decay. A more complete history of singing-voice synthesis can be found in [7].

2.2 Formant-Wave-Function Synthesis

Developed in the 1980s by Xavier Rodet at IRCAM in Paris, the original FOF synthesis technique [1] involved determining the time-domain response $y(n) = (x * h)(n)$ of a filter with impulse response $h(n)$ to a particular excitation signal $x(n)$. Assuming that the excitation signal is a periodic repetition of impulses or other such excitation shape, the time-domain representation of the output of the filter can be determined and the synthesis can be performed in the time-domain by repeating the output signal at the period of a desired fundamental frequency and performing an overlap–add operation to obtain a single output stream.

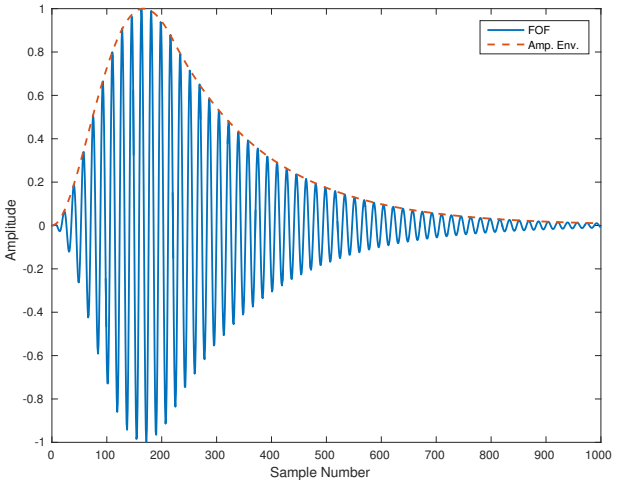


Figure 1. FOF with $\alpha = 80\pi$ rad/s, $\pi/\beta = 5$ ms, $\omega_c = 2500$ rad/s and SR = 44.1 kHz

The following time-domain function was preferred [1]:

$$y(n) = \frac{1}{2}\left(1 - \cos[\beta nT]\right) e^{-\alpha nT} \sin\left(\omega_c nT + \phi\right) \quad \text{for } 0 \leq n \leq k, \quad (1)$$
$$y(n) = e^{-\alpha nT} \sin\left(\omega_c nT + \phi\right) \quad \text{for } n > k,$$

where $k = \pi/(\beta T)$ controls the amplitude envelope attack time as well as the skirtwidth of the formant region in the frequency domain, α controls the bandwidth of the formant region and ω_c controls the center frequency of the formant region. The term skirtwidth refers to the width of the formant region measured further away from the peak than the bandwidth is measured. In other words, this function generates an exponentially decaying sinusoid whose initial discontinuity is smoothed over k samples (Figure 1).

The form of Eq. (1), while expressed in the time domain, was chosen for its spectral properties. The bandwidth and skirtwidth can be controlled independently of each other and the spectral envelope of the formant region is symmetric about the center frequency.

By running multiple FOF generators in parallel and summing the outputs it is possible to generate a wide variety of vocal and instrumental sounds using this method. In particular, this method is used in the software program CHANT developed at IRCAM in the 1980s [2]. The original CHANT program was implemented at IRCAM using an FPS-100 array processor and was later ported to C in the early 1990s so that CHANT could run on any Unix or Macintosh system [8].

In the late 1980s, a version of FOF was ported to Music 11 as part of the VOCEL (VOiCe ELeven) project by Michael Clarke [9]. This version of FOF included additional features such as allowing different fundamental frequencies for individual FOF generators and an octaviation effect. Additionally, this implementation used a lookup table containing a user-specified attack shape which determined both the attack and decay envelope of a single FOF burst. This version of FOF was later ported to Csound [10].

A different approach was taken by Philippe Depalle et al. in 1992 [11]. Their approach was to separate the excitation signal from the resonant filter by developing a time-domain function for the excitation signal implied by Eq. (1). Then, the excitation signal could be run through a simple second-order resonant bandpass filter. This would allow utilization of the filtering schemes on DSP chips.

A nice property of this scheme is that it was no longer necessary to overlap-add time-domain functions nor worry about audible noise created by truncating the time-domain FOF before it has sufficiently decayed. Additionally, [11] provides a compact function for controlling the skirtwidth of the formant region created by the frequency response of the filter output.

An approach suitable for VLSI implementation on a DSP chip was developed in 1996 by J. Spanier et al. [12]. Their method involved developing a filter that would generate an amplitude envelope with favorable spectral properties. The filter could then be excited by an impulse train and the output of the filter used to envelope the output of a sinusoidal oscillator.

More recently, Michael Clarke and Xavier Rodet ported FOF synthesis to Max [13]. This version contained features of both the FOF version contained in Csound and early FOF objects already in Max.

SuperCollider has a uGen `formlet` [14] dating back to 2002 [1] which forms a FOF-type wave burst using the difference between two second-order resonant bandpass filters having different decay rates but the same center frequency. Its amplitude envelope is given by the uGen `decay2` [15] which uses the difference of two one-pole exponential decays to create an attack-smoothed exponential envelope.

3. HYBRID FILTER–OSCILLATOR FOF IMPLEMENTATION

In this paper a hybrid filter-wavetable oscillator model is employed in which an impulse train is fed into a second-order filter which generates a FOF amplitude envelope similar to the amplitude envelope in Figure 1. That envelope is then multiplied by a sinusoidal wavetable oscillator to generate the FOF waveform.

3.1 Filter Derivation

As seen in Section 2, previous hybrid FOF implementations have either found an explicit formula for a smooth excitation filter to feed into a resonant bandpass filter or designed a modified filter with a smoother attack that can be excited with a periodic impulse train. One such filter, which implements the amplitude envelope $t^2 e^{-\alpha t}$, is mentioned in [12] but was not used as the authors found the frequency response to be unsuitable for FOF synthesis. Additionally, that filter does not allow for control of the skirtwidth of the generated formant region.

As shown in [16], in the context of artificial reverb generation, the convolution of two decaying exponential envelopes (Figure 2) yields an envelope with a smoothed at-

[1] James McCartney, personal and email communication, Apr. 22-23, 2016

tack that has a shape similar to other FOF envelopes (such as the one generated by Eq. (1) in Figure 1).

While comb filters with controllable feedback delay-time parameters are used in the setting of artificial reverberation, for the generation of a FOF amplitude envelope, one-pole resonators with unit delays are sufficient:

$$\hat{H}(z) = \frac{1}{1-\mu z^{-1}}, \tag{2}$$

where $\mu = e^{-\alpha T}$ is the coefficient of decay, T is the sampling period and $\alpha = 1/\tau$ is the inverse of the decay time-constant τ. This filter has impulse response

$$\hat{h}(n) = \mu^n u(n), \tag{3}$$

where $u(n)$ is the unit step, $u(n) = 1, n \geq 0$, and $u(n) = 0, n < 0$. Thus, the convolution of two such one-pole filters

$$h(n) = \hat{h}_1(n) * \hat{h}_2(n) = \mu_1^n * \mu_2^n = e^{-\alpha_1 nT} * e^{-\alpha_2 nT} \tag{4}$$

corresponds to the multiplication of their transfer functions, giving the following second-order filter

$$\begin{aligned} H(z) &= \frac{1}{1-e^{-\alpha_1 T}z^{-1}} \cdot \frac{1}{1-e^{-\alpha_2 T}z^{-1}} \\ &= \frac{1}{(1-\mu_1 z^{-1})(1-\mu_2 z^{-1})} \\ &= \frac{1}{1-(\mu_1+\mu_2)z^{-1}+\mu_1\mu_2 z^{-2}}. \end{aligned} \tag{5}$$

This filter will produce an amplitude envelope in the time-domain whose frequency response is a formant centered at dc (0 Hz). Assuming that $\alpha_1 > \alpha_2$ so that $\mu_1 < \mu_2$, $\alpha_2 = \pi \cdot \text{BW}$ can be used to control the decay time of the amplitude envelope which will in turn control the -3 dB bandwidth of the formant (where BW is the bandwidth in Hz).

The other filter parameter α_1 can be used to tune the rise time of the amplitude envelope which will also control the skirtwidth of the formant region. For a constant value of μ_2, as $\alpha_1 \to \alpha_2$ it follows that $\mu_1 \to \mu_2$. When $\mu_1 = \mu_2$, the longest rise time is achieved which gives the narrowest possible skirtwidth in the frequency response for the bandwidth controlled by α_2. Conversely, as $\alpha_1 \to \infty$, $\mu_1 \to 0$ so the attack time becomes shorter with the filter becoming a one-pole exponential decay in the limit.

The impulse response of the amplitude envelope is given by [16] to be

$$h(n) = \frac{1}{\mu_2 - \mu_1}\left(\mu_2^{n+1} - \mu_1^{n+1}\right). \tag{6}$$

The rise time—the time of the impulse response maximum—may be found as the time in at which the impulse response time derivative

$$h'(n) = \frac{T}{\mu_2 - \mu_1}\left(\alpha_1\mu_1^{n+1} - \alpha_2\mu_2^{n+1}\right) \tag{7}$$

is zero. Expressing the time in seconds, we have

$$nT = \frac{\ln(\alpha_1/\alpha_2)}{\alpha_1 - \alpha_2} - T. \tag{8}$$

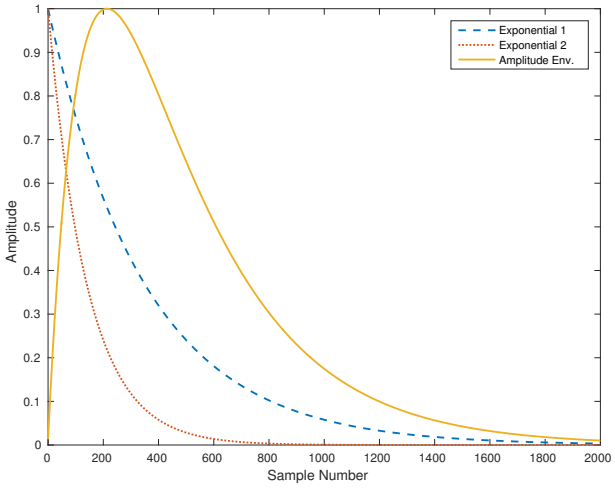


Figure 2. Amplitude envelope generated by the convolution of two decaying exponentials

Additionally, the maximum allowable rise time for a particular bandwidth α_2 can be determined by taking the limit of Eq. (8) as $\alpha_1 \to \alpha_2$. In detail, making use of L'Hôpital's rule, it is found that

$$\begin{aligned} nT_{max} &= \lim_{\alpha_1 \to \alpha_2} \frac{\ln(\alpha_1/\alpha_2)}{\alpha_1 - \alpha_2} - T = \lim_{\alpha_1 \to \alpha_2} \frac{1}{\alpha_1} - T \\ &= \frac{1}{\alpha_2} - T. \end{aligned} \tag{9}$$

In order to calculate an α_1 that satisfies a particular choice of α_2 and rise time $nT \leq nT_{max}$, it is necessary to solve Eq. (8) iteratively or using the Lambert W function [17]. If Eq. (8) is put in the form $Y = Xe^X$, then $X = \mathcal{W}(Y)$ where $\mathcal{W}$ denotes the Lambert W function. Rearranging Eq. (8) so that it is in the prerequisite form and then applying the Lambert W function gives

$$\alpha_1 = \frac{-1}{(n+1)T}\mathcal{W}\left(-\alpha_2(n+1)Te^{-\alpha_2(n+1)T}\right). \tag{10}$$

3.2 Filter Amplitude Normalization

The filter developed in Section 3.1 has the largest amplitude response at dc. Therefore, normalization is achieved by normalizing the dc amplitude response of the filter. The frequency response of the filter is given by:

$$H(e^{j\omega T}) = \frac{1}{1-(\mu_1+\mu_2)e^{-j\omega T}+\mu_1\mu_2 e^{-2j\omega T}} \tag{11}$$

and so the magnitude response at dc is given by:

$$\begin{aligned} G(0) = |H(0)| &= \left|\frac{1}{1-(\mu_1+\mu_2)+\mu_1\mu_2}\right| \\ &= \frac{1}{(1-\mu_1)(1-\mu_2)}. \end{aligned} \tag{12}$$

Thus, the gain coefficient $\gamma = 1/G(0)$ will normalize the magnitude response of the filter to 0 dB. Then, any other magnitude level is easily obtained by applying the correct linear scale factor $g = 10^{\beta/20}$ where β dB is the desired offset in dB.

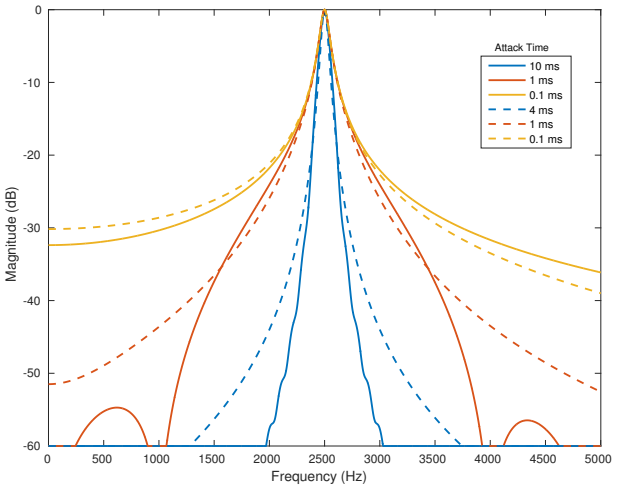


Figure 3. Comparison of frequency characteristics between original and proposed FOF with: $\alpha = 80\pi$ rad/s, $\omega_c = 2500$ rad/s and varying rise times (Orig.: solid lines, Proposed: dashed lines)

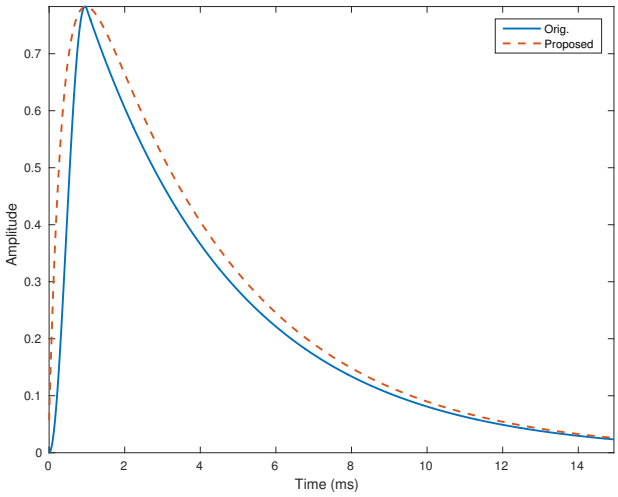


Figure 4. Comparison of amplitude envelope characteristics between original and proposed FOF with: $\alpha = 80\pi$ rad/s, $\omega_c = 2500$ rad/s and 1 ms rise times

3.3 Complete FOF Architecture

The complete FOF structure is comprised of an impulse train that triggers the amplitude envelope filter which is subsequently multiplied by a sinusoidal oscillator. The frequency f_0 of the impulse train determines the fundamental frequency of the synthesized sound and the sinusoidal oscillator determines the center frequency f_c of the formant region created by the FOF.

The filter defined in Eq. (5) can be expanded using partial fraction expansion into a difference of one pole filters similar to `decay2` [15] in SuperCollider. The sinusoidal oscillator can be combined with Eq. (5) transforming it into a fourth-order resonant bandpass filter which can similarly be expanded into a difference of second-order resonant bandpass filters like `formlet` [14]. The performance differences between the implementations will be touched upon in Section 5.

A comparison of the spectral qualities of the method in this paper with Rodet's original FOF technique (Eq. (1)) is illustrated in Figure 3. A single decay rate with a variety of rise times are plotted. Overall, the bandwidths are quite comparable between both techniques with the main difference being that the original FOF formula can achieve considerably narrower bandwidths for a fixed decay rate. For the 80 Hz bandwidth used in the example, our technique was not able to match the 10 ms rise time and reached an upper bound of approximately 4 ms. The biggest difference, as seen in Figure 3, is the narrowness of the skirtwidth achieved by the original FOF technique. The authors did not investigate whether or not this difference is perceptible. While the shapes of the skirts are slightly different, it is clear from the figure that independent skirtwidth control is provided by the proposed technique.

Figure 4 shows the difference in the amplitude envelopes between Rodet's technique and the proposed technique. The attack times match exactly but the rise and decay shapes are slightly different which makes sense given the spectral differences observed in Figure 3.

3.4 Comparison to SuperCollider

A particularly notable feature of this implementation comes from the design of the FOF envelope filter. If the bandwidth and rise parameters `bw` and `a` are set equal to each other, which causes the poles μ_1 and μ_2 to be equal, the filter in Eq. (5) becomes a second order filter with repeated poles. Additionally, for `bw` $<$ `a`, the roles of the parameters are just reversed so that `a` controls the decay time and bandwidth whereas `bw` controls the rise time and skirtwidth.

This is in contrast to the behavior of the `decay2` and `formlet` uGens in SuperCollider. With `bw` $=$ `a`, both objects produce a constant signal of zero and when `a` $<$ `bw`, `decay2` produces an inverted envelope and `formlet` similarly produces an inverted FOF wave burst.

4. FAUST IMPLEMENTATION

The FOF filter structure developed in Section 3.1 was implemented using the FAUST (*Functional Audio Stream*) programming language [18]. FAUST allows for the quick prototyping and development of DSP algorithms with short, succinct lines of code. The FAUST code compiles down to C++ code that can then ported to standalone applications, externals for a variety of other computer music languages or embedded within a larger C++ project.

4.1 A Basic FOF Generator

A single FOF generator consists of an amplitude envelope, which is generated using a second-order filter, multiplied by a sinusoidal oscillator. In order for good quality synthesis using this configuration, the sinusoidal oscillator must be hard-synced to the same starting phase at the beginning of each new amplitude envelope. At the time of this writing, the FAUST libraries did not contain a hard-syncing

oscillator, so the wavetable oscillator `osc` found in the library `music.lib` was modified to include that functionality (Listing 1).

Listing 1. Hard-Syncing Wavetable Oscillator

```
// import FAUST music library
ml = library("music.lib");

// resettable phasor, clock val > 0 resets phase to 0
ph(f0,c) = inc : (+ : d)~ (-(_<:(_,*(_,clk)))) : *(ts)
with {
        clk = c>0;
        d = ml.decimal;
        inc = f0/float(ml.samplingfreq);
        ts = float(ml.tablesize);
      };

// sin lookup table with resettable phase
oscpr(f0,c) = rdtable(ml.tablesize, ml.sinwaveform,
                      int(ph(f0,c)));
```

In particular, the phasor that controls table-lookup was modified to reset the phase to zero every time a non-zero clock signal is received.

Next is the code for a FOF generator. The generator takes four parameters:

- `fc`: the center frequency in Hz of the formant region
- `bw`: the bandwidth in Hz which controls the decay rate of the amplitude envelope
- `a`: the rise-time bandwidth in Hz which controls the rise time of the amplitude envelope
- `g`: a linear gain factor where `g` $= 1$ corresponds to a 0 dB peak frequency response

Additionally, the FOF signal block must be connected to a clock signal which should impulse the FOF generator at the desired fundamental frequency. The corresponding FAUST code for the FOF generator is given in Listing 2 and the block diagram in Figure 5.

Listing 2. FOF Generation System

```
// import FAUST filter and music libraries
fl = library("filter.lib");
ml = library("music.lib");

// function to generate a single Formant-Wave-Function
fof(fc,bw,a,g) = _ <: (_',_) :(f * s) with {
  T  = 1/ml.SR;                // sampling period
  pi = ml.PI;
  u1 = exp(-a*pi*T);
  u2 = exp(-bw*pi*T);
  a1 = -1*(u1+u2);
  a2 = u1*u2;
  G0 = 1/(1+a1+a2);            // dc magnitude response
  b0 = g/G0;                   // normalized filter gain
  s  = oscpr(fc);              // wavetable oscillator
  f  = fl.tf2(b0,0,0,a1,a2); // biquad filter
};
```

Taking the code from Listing 1 and Listing 2, it is possible to generate high quality vocal synthesis. However, it is necessary that the formant regions, as controlled by the bandwidth and rise-time parameters, are held constant or varied slowly over time so that audible artifacts are not introduced by time-varying the filter coefficients.

4.2 Filter Cycling and Coefficient Management

Since the FOF bursts typically overlap but the filter coefficients need to remain fixed during the audible duration of a single FOF burst, it is desirable to develop a more robust control structure to allow for the realtime manipulation of the FOF envelope parameters. This robustness can be achieved by introducing filter cycling and a sample-and-hold mechanism on the decay and rise filter coefficients.

Listing 3. Cyclic Impulse Train Streams

```
// import the oscillator library
ol = library("oscillator.lib");

// impulse train at frequency f0
clk(f0) = (1-1')+ol.lf_imptrain(f0)';
// impulse train at frequency f0 split into n cycles
clkCycle(n,f0) = clk(f0) <: par(i,n,resetCtr(n,(i+1)));
// function that lets through the mth impulse out of
// each consecutive group of n impulses
resetCtr(n,m) = _ <: (_,ctr(n)) : (_,(_==m)) : *;
// function to count nonzero inputs and reset after
// receiving x of them
ctr(x) = (+(_)~(negSub(x)));
// function that subtracts value x from
// input stream value if input stream value >= x
negSub(x)= _<: (_>=x,_,_):((-1*_),_,_):((_*_),_):(_+_);
```

With filter cycling, n identical filters are implemented in parallel and the impulse train is distributed so that the jth impulse is routed to the $[(j \bmod n) + 1]$th filter. Thus, if the filter coefficients are held constant using a sample-and-hold mechanism until the next impulse is received and enough filters are used so that each filter has sufficiently decayed in amplitude prior to its next impulse arriving, the filter coefficients can be swept as quickly as desired with no audible artifacts.

To handle the distribution of the impulses to the n different filters in FAUST a resetting series of counting mechanisms were developed (Listing 3, Figure 6). The main function in the listing is `clkCycle` which cyclically distributes unit gain impulses among n different output streams. The function `clk` outputs a single unit gain impulse stream at the frequency provided in the input argument. It is a slight modification of the function `lf_imptrain` included in FAUST's `oscillator.lib` library that compensates for a one period delay in the arrival of the first impulse. The

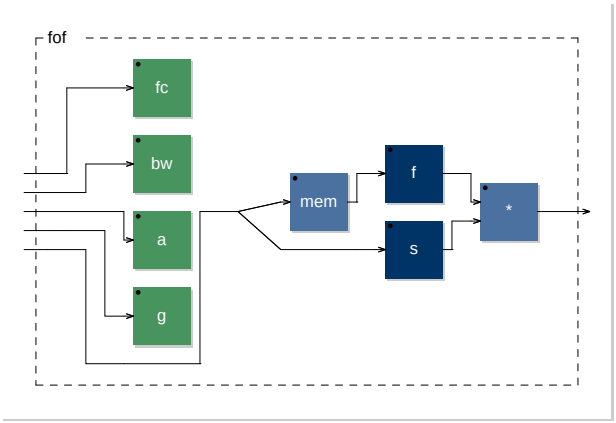


Figure 5. Block diagram of the code from Listing 2

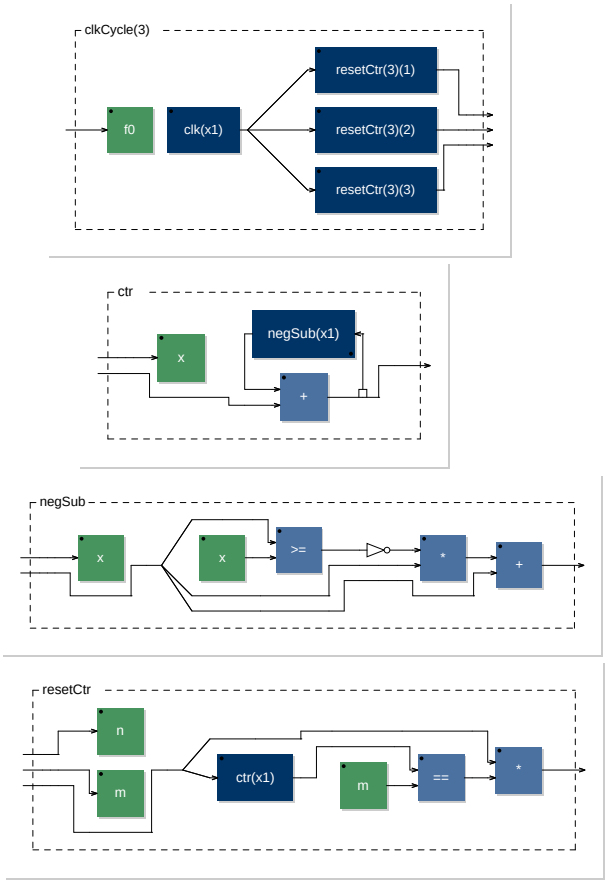


Figure 6. Block diagrams corresponding to Listing 3

functions `ctr`, `resetCtr` and `negSub` work in tandem to accomplish the internal management that `clkCycle` needs to function correctly.

With the distribution of the impulse train accomplished, it is then necessary to implement the sample and hold mechanism. The `filter.lib` library provides a sample-and-hold filter called `latch` which takes two signals: an input signal and a clock signal. It samples the input signal every time the clock signal goes nonnegative and always outputs the currently held value. Thus, to implement sample and hold, it is just necessary to connect the relevant parameter signals to `latch` then feed `clkCycle` to `latch` and feed a one-sample delay of `clkCycle` to `fof` so that the sampling occurs just before the filter is excited. The code and block diagram for the sample-and-hold mechanism and the corresponding FOF implementation are provided in Listing 4 and Figure 7.

Listing 4. FOF with Sample-and-Hold

```
// import the filter library
fl = library("filter.lib");

// sample and hold filter coefficients
curbw = (_,bw) : fl.latch;
cura = (_,a) : fl.latch;

// FOF sample and hold mechanism
fofSH = _ <: (curbw,cura,_) : (fc,_,_,g,_') : fof;
```

With the sample-and-hold mechanism in place, all parameters can be mapped from GUI elements in a standalone or DAW plugin or can be manipulated via control signals in another computer music language.

Listing 5. Example Program

```
/************** parameters/GUI controls **************/
// fundamental freq (in Hz)
f0 = vslider(``F0'',220,0,2000,0.01);
// formant center freq (in Hz)
fc = vslider(``Fc'',800,100,6000,0.01);
// FOF filter gain (in dB)
g = ml.db2linear(vslider(``Gain'',0,-40,40,0.01));
// FOF bandwidth (in Hz)
bw = vslider(``BW'',80,1,10000,1);
// FOF attack value (in Hz)
a = vslider(``A'',90,1,10000,1);

// number of S&H cycling filters
n = 5;

// main process
process = clkCycle(n,f0) <: par(i,n,fofSH) :> _;
```

The code in Listings 1–4 is all that is necessary to perform FOF synthesis using FAUST. A simple complete example FAUST program is provided in Listing 5. The program generates a single FOF wave stream with the bandwidth, rise time, center frequency, gain and fundamental frequency values controlled externally or by GUI elements. The program features five identical FOF streams running cyclically in parallel. Keep in mind that the code from Listings 1–4 are not being included in example program listing for brevity's sake but would need to be included in the actual program in order for it to compile and run.

5. CONCLUSIONS

In this paper a system for FOF synthesis was presented that uses a hybrid filter–wavetable oscillator architecture. The convolution of two exponential decay one-pole filters is used to generate an exponentially decaying amplitude envelope with smoothed attack. The amplitude envelope is then multiplied by a hard-synced wavetable sinusoid generator to generate FOF wave bursts which can then be used for FOF synthesis. The proposed technique was implemented in the FAUST audio programming language. Finally, filter cycling and sample-and-hold mechanisms were

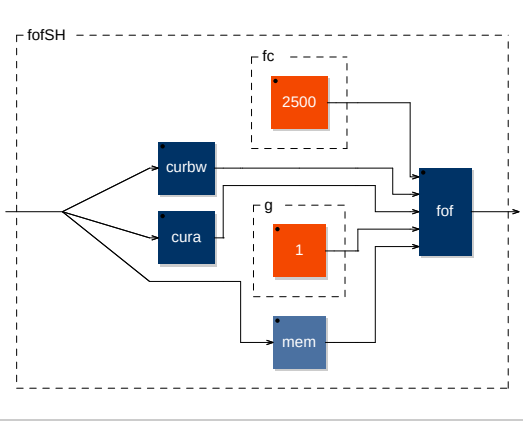


Figure 7. Block diagram corresponding to Listing 4

added to improve the robustness and flexibility of the synthesis technique.

A potential next stage in this research would be to develop a gain correction computer based on the fundamental frequency of the synthesized tone. The gain computer would be used to automatically normalize the increase in energy and, hence, volume that occurs when the fundamental frequency is increased which causes an increase in the overlapping between each successive FOF wave burst.

It would also be of potential interest to develop an alternative FOF envelope filter that does not suffer from the limitation of the maximum rise time being coupled to the value of the bandwidth parameter. That would lead to a hybrid technique capable of producing the extremely narrow skirtwidths possible with the original time-domain FOF synthesis technique.

Finally, some of the features of the Csound FOF implementation [10] such as octaviation, controlling different FOF streams with different fundamental frequency clock signals and using the FOF envelope with signals other than pure sinusoids could be added to the proposed implementation.

Acknowledgments

Michael Jørgen Olsen would like to acknowledge fruitful and insightful conversations with Romain Michon, Elliot Kermit-Canfield, Paul Batchelor and Maximilian Rest.

6. REFERENCES

[1] X. Rodet, "Time-domain formant-wave-function synthesis," Computer Music Journal, vol. 8, no. 3, pp. 9–14, 1984.

[2] X. Rodet, Y. Potard, and J. B. Barrière, "The CHANT project: From the synthesis of the singing voice to synthesis in general," Computer Music Journal, vol. 8, no. 3, pp. 15–31, 1984.

[3] M. R. Schroeder, "Vocoders: Analysis and synthesis of speech (a review of 30 years of applied speech research)," Proc. IEEE, vol. 54, pp. 720–734, May 1966.

[4] J. L. Kelly and C. C. Lochbaum, "Speech synthesis," Proc. Fourth Int. Congress on Acoustics, Copenhagen, pp. 1–4, September 1962.

[5] L. R. Rabiner, "Digital-formant synthesizer for speech-synthesis studies," The Journal of the Acoustical Society of America, vol. 43, no. 4, pp. 822–828, 1968.

[6] D. H. Klatt, "Software for a cascade/parallel formant synthesizer," The Journal of the Acoustical Society of America, vol. 67, no. 3, pp. 971–995, 1980.

[7] P. R. Cook, "Singing voice synthesis: History, current work, and future directions," Computer Music Journal, vol. 20, no. 3, pp. 38–46, 1996.

[8] J. B. Barrière, F. Iovino, and M. Laurson, "A new CHANT synthesizer in C and its control environment in PATCHWORK," in Proc. Intl. Computer Music Conf., Montréal, Canada, Oct. 16–20 1991, pp. 11–14.

[9] J. M. Clarke, P. Manning, R. Berry, and A. Purvis, "VOCEL new implementations of the FOF synthesis method," in Proc. Intl. Computer Music Conf., Kologne, Germany, Sept. 20–25 1988, pp. 357–371.

[10] J. M. Clarke, "FOF and FOG synthesis in Csound," in The Csound book: Perspectives in software synthesis, sound design, signal processing and programming, R. Boulanger, Ed. MIT Press, 2000, pp. 293–306.

[11] P. Depalle, D. Matignon, and M. Stroppa, "Source-filter formulation and analytic control of the skirtwidth of CHANT formant-wave-functions," in Proc. Intl. Computer Music Conf., San Jose, USA, Oct. 14–18 1992, pp. 372–373.

[12] J. R. Spanier, S. Johnson, and A. Purvis, "Optimisations of the FOF algorithm for VLSI implementation," in Proc. Intl. Computer Music Conf., Hong Kong, Aug. 19–24 1996, pp. 493–495.

[13] J. M. Clarke and X. Rodet, "Real-time FOF and FOG synthesis in MSP and its integration with PSOLA," in Proc. Intl. Computer Music Conf., Singapore, Sept. 29–Oct. 4 2003.

[14] J. McCartney, "Formlet," 1996. [Online]. Available: http://doc.sccode.org/Classes/Formlet.html

[15] ——, "Decay2," 1996. [Online]. Available: http://doc.sccode.org/Classes/Decay2.html

[16] K. Lee and J. Abel, "A reverberator with two-stage decay and onset time controls," in Proc. Audio Eng. Soc. (AES) Conv., vol. 129, San Francisco, CA, Nov. 4–7 2010.

[17] R. M. Corless, G. H. Gonnet, D. E. G. Hare, D. J. Jeffrey, and D. E. Knuth, "On the Lambert W function," Adv. Comput. Math., vol. 5, no. 4, pp. 329–359, 1996.

[18] Y. Orlarey, "Faust," 2002. [Online]. Available: http://faust.grame.fr/

THE PERCEPTUAL SIMILARITY OF TONE CLUSTERS: AN EXPERIMENTAL APPROACH TO THE LISTENING OF AVANT-GARDE MUSIC

Arvid Ong **Reinhard Kopiez**

Hanover University of Music, Drama and Media, Hanover, Germany

arvid.ong1@hanse.net reinhard.kopiez@hmtm-hannover.de

ABSTRACT

This study examines the musical tone cluster as a prototypical sound of avant-garde music in the 20th and 21st century. Tone clusters marked a turning point from pitch-related techniques of composing in earlier epochs to the sound-based materials used in avant-garde music. Henry Cowell offered the first theoretical reflection about the structure of clusters with a focus on tone density which relying on the number of tones and the ambitus of the cluster. In this paper we first show the results of a sound discrimination experiment, where participants had to rate the sound similarity of prototypical cluster sounds with varying in densities. The results show congruency between theoretical features of the cluster structure, results of the timbre feature analysis, and perceptual evaluation of stimuli. The correlation between tone cluster density and psychoacoustical roughness was r = .95 and between roughness and similarity ratings r = .74. Overall, the similarity ratings of cluster sounds can be grouped into two classes of sounds: (a) those clusters with a high grade of perceptual discrimination depending on the cluster density and (b) those clusters of a more aurally saturated structure making it difficult to separate and evaluate them. Additionally, the relationship between similarity ratings and psychoacoustical features was also examined. Our findings can provide valuable insights into aural training methods for avant-garde music.

1. INTRODUCTION

Since the beginnings of avant-garde music in the 20th century, the aspect of "sound" has become the essential issue in compostional techniques. In the early 1910s, the American composer Henry Cowell (1897-1965) founded a new theoretical conception of sound through the exploration of tone clusters. In addition, Cowell developed a basic theory about tone clusters in his book *New Musical Resources* [1]. Tone clusters marked a turning point in music history: they make up chords which are characterized by noisiness in opposite to "tonalness, that describes the hearing-related characteristic how a sound differs from noise." [2, p. 363] Therefore, the categories of tonal harmony, i. e. consonance and dissonance, became less sufficient for understanding harmonic theory. In 1960, György Ligeti captured this idea of tone cluster sounds in his orchestral piece *Atmosphères* prominently employing this innovative compositional technique. This led to the musicological need for the development of a compositional theory about the elements of musical sounds.

It was the aim of this research to get a first impression about the perception of tone clusters as a principal prototype of the modern chord. In earlier research, music with tone clusters was interpreted through graphical analysis of the music score. However, in this study we have considered methods of music psychology that allow us to investigate more precisely the perceptual aspects associated with avant-garde music. In addition to our psychological study, we performed a psychoacoustical feature analysis on the stimulus set.

The dependent variables for cluster evaluation used in this study were *interestingness* [3] and *similarity*. We used the participants' perception of similarity as our basic psychological concept for listening to music. Our first hypothesis is based on the influence of *tone cluster density*, which describes the relation between the number of individual tones and the overall range of the tone cluster. Figure 1 shows a hypothetical graph of the relation between the rating value of the perception of similarity and tone cluster density under two hypothesized effects. We assumed that the similarity of two tone clusters is characterized by a global maximum distributed around the difference of densities ΔD = 0 (hypothesis A). Otherwise, if density had only a small effect, it would be assumed that the perception of similarity is determined by a saturation effect with increasing density (hypothesis B).

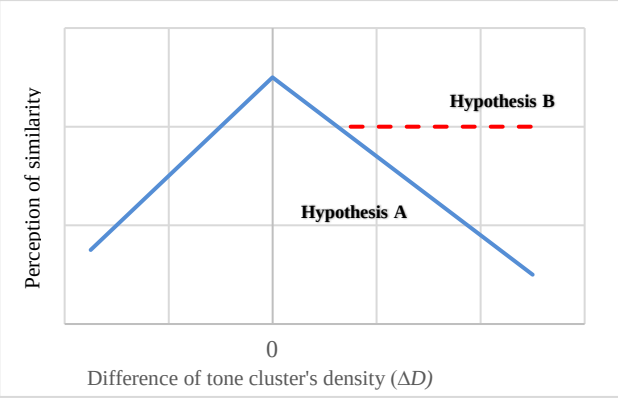


Figure 1. Hypothesized perception of similarity.

The tone clusters' height and range of all stimuli have been fixed in the experimental design to remove their influence on the perception of similarity.

2. THEORY

2.1 Definition

Tone clusters are musical chords formed by three or more adjacent tones that usually come from narrowly spaced intervals: for instance, seconds in twelve tone equal-temperament. However, pieces using microtonal intervals exist as well (e.g., Krzysztof Pendereckis *Anaklasis* uses quarter-tones). The noisiness of a sound is a specific property of tone clusters.

2.2 Cowell's theory of tone clusters

Henry Cowell gave the following definition: "Tone-clusters, then, are chords built from major and minor seconds, which in turn are derived from the upper reaches of the overtone series and have, therefore, a sound foundation." [1, p. 117] Cowell called the grouping of three tones spaced in seconds *cluster-triads*. He compared the chord formation of tone clusters with the harmonic principles of tonal music: Just as there are combinations of major and minor thirds in conventional triads in tonal music, there are combinations of major and minor seconds in cluster-triads (see Figure 2). Larger tone clusters are combinations of these basic elements.

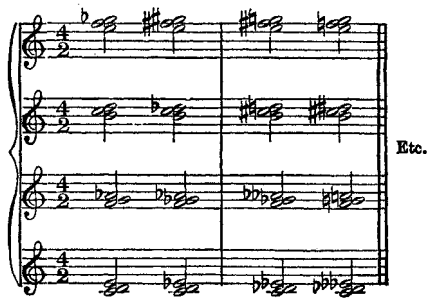


Figure 2. Cowell's example of tone cluster-triads [1, p. 117].

In this way Cowell describes the inner structure of tone-clusters: "These four triads are the basis of all larger clusters, which can have great variety, owing to the many different possible juxtapositions of the triads within larger clusters." [1, p. 118] Cowell also described tone clusters related to the layout of the keyboard. Tone clusters played on the black keys corresponded to the pentatonic scale and those played on the white keys to the diatonic scale. This shows that Cowell's theoretical approach was mostly informed by intuition, although he argues that tone clusters and cluster-triads are related to the higher reaches of the overtone series. In this way, Cowell depicted tone clusters as an inevitable progression in the development of music in history.

2.3 Density of tone clusters

The substantial idea of tone clusters is the combination of a large number of seconds in the chord. Clearly, it becomes impossible to distinguish single pitches or intervals in a tone cluster which is the reason that density is considered as an alternative theoretical explanation. The term "density" was first proposed by Mauricio Kagel as a specification of Cowell's cluster-triads [4].

The density of tone clusters relates to the number of pitches contained in the chord and its ambitus. It is calculated by the equation

$$D = \frac{N-1}{W} \cdot 12 \qquad (1)$$

where *D* is the density, *N* the quantity of tones, *W* the width of the tone cluster and 12 is a normalization factor. Unlike its common usage in traditional harmony, octave equivalency is not considered in this theoretical approach. This means that it is necessary to count all registers of the pitches of the chords as well as those in octaves, which are usually ignored (e. g. in A. Fortes "pitch-class set theory", see [5], see also [6]).

3. EXPERIMENTAL METHOD

3.1 Stimuli

For this study, we chose ten tone cluster sounds of different densities which have the same ambitus (see Figure 3).

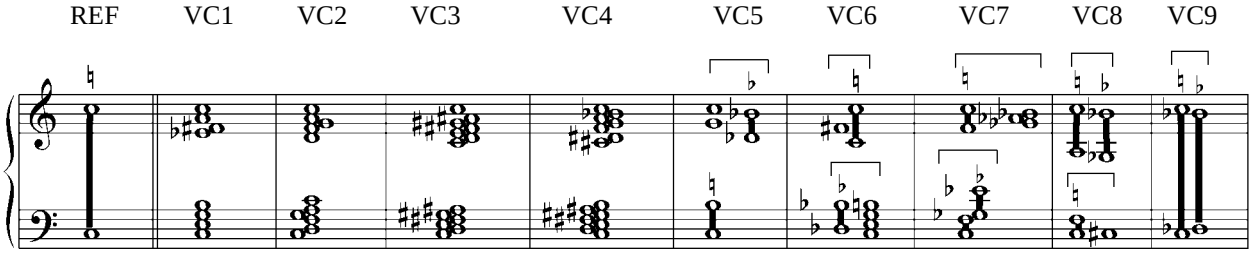


Figure 3. Score of the ten stimuli used in the experiment. These bars indicate a tone-cluster ranging over the two notes outlined. A flat over the tone-cluster indicates that all black keys are used, a natural that all white keys are used.

The stimuli were produced by a MIDI-controlled sample library piano sound. The range of all tone clusters was two

octaves or $W = 24$ semi-tones. The pitch range was fixed from C3 to C5 (181.1 to 523.3 Hz). All stimuli had a duration of 3.5 s. One of the ten tone clusters was chosen as a reference sound. This reference chord can be described as a combination of tones based on the diatonic scale. This cluster can be regarded as prototypical and is found in Cowell's piano works, for example. The cluster density of the reference item has a mean value of $D_{REF} = 7.0$ tones per octave.

The other items were designed to represent various grades of density from 3.5 to 12.0 (tones per octave, see Table 1).

Cluster	Density *D*	Number of tones *N*
REF	7.0	15
VC1	3.5	8
VC2	5.0	11
VC3	6.0	13
VC4	6.5	14
VC5	7.0	15
VC6	8.5	18
VC7	9.5	20
VC8	10.5	22
VC9	12.0	25

Table 1. Densities and the number of tones from the cluster stimuli REF and VC1-VC9.

Some chords are typically used in works with tone clusters: Item VC2 is based on the pentatonic scale and item VC3 on the whole tone scale. Item VC5 is a combination of two different diatonic tone clusters that are non-equivalent in both octaves. The tone cluster items VC6-VC8 were constructed with intervallic structures which were composed arbitrarily. For these chords with densities between 8.0 and 10.5 there are no specific references within music theory literature. Thus, these items completed the selection of stimuli with those grades of density which were not represented through the other tone cluster stimuli.

3.2 Participants

3.2.1 Demographic Data

$N = 50$ students of the Hanover University of Music, Drama and Media (26 female) took part in the experiment. The mean age was $M = 21.9$ years, $SD = 2.6$ (range between 19 and 29 years).

3.2.2 Musical experience

The participants were students in music performance, music education or musicology. The students had been studying for varying durations of between 1 and 15 semesters. Most of them played one or more instruments (16 pianists, 11 singers, 10 guitarists or string players. Additional instruments were percussion, trumpet, flute or saxophone).
As a control variable, the degree of "musical sophistication" of the participants was determined by the German version of the Goldsmiths Musical Sophistication Index [7]. The inventory's "general factor" (18 items) and "musical training" (1 additional item) were applied. Participants were asked to indicate their degree of agreement for each statement on a seven-point Likert scale. They completed the questionnaire before the auditory experiment.

3.3 Design and Procedure

There were two dependent variables in the experimental design of this study: *interestingness* and *similarity* of each sound when compared to the reference cluster. The experiment was divided into two parts. The operationalization of the experiment was supported by a customized version of the STEP software package [8]. The two computer displays that were visible to the participants during the experiment are shown in Figure 4 and Figure 5.
In the first part, the participants were asked to rate the ten tone clusters. The following instruction was given: "Please rate the ten sounds 'A' to 'J' on a scale from 0 'very uninteresting' to 100 'very interesting'" (see Figure 4). In this phase of the experiment, the participants were not informed about the reference cluster. Participants could repeatedly listen to the sounds and compare them directly. This procedure also served to familiarize participants with the stimuli.
The second part of the experiment was an adaption of the Multi Stimulus with Hidden Reference and Anchor (MUSHRA) paradigm, reported in a recommendation of the International Telecommunication Union [9]. In this design, the reference tone cluster REF is used as an anchor. Participants were asked to compare the reference item with each of the other nine stimuli (VC1 to VC9). The presented stimuli were computer-controlled with difference randomization for each participant. The participants did not know which tone cluster was being presented. The panel displayed the following instruction: "Please rate the similarity between sound 'A' and sound 'REF' on a scale from 0 'maximally dissimilar' to 100 'maximally similar'" (see Figure 5).

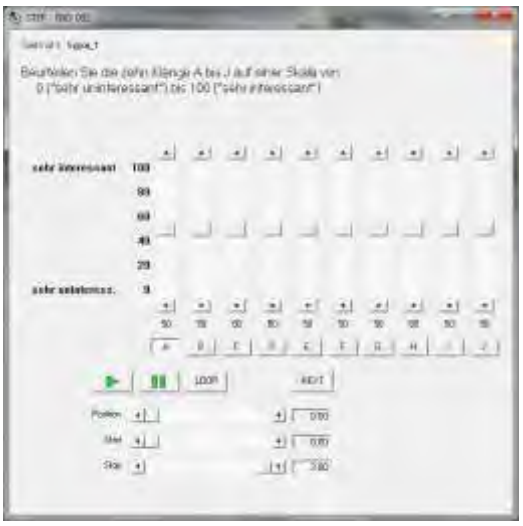


Figure 4. Panel from the first part (interestingness rating): simultaneous presentation of the ten stimuli.

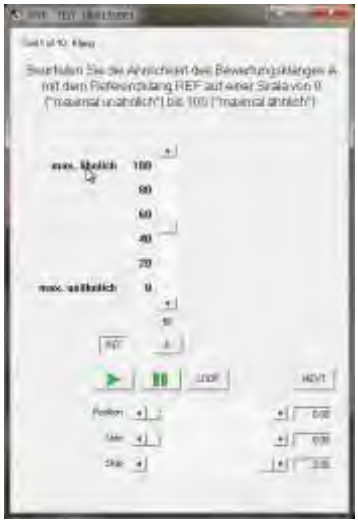


Figure 5. Panel of the second part (similarity rating): pairwise comparison with the reference stimulus.

4. RESULTS

4.1 Musical sophistication index of participants

The mean Gold-MSI general factor score of the sample was *M* = 99.1 [75, 112], *SD* = 8.5. The maximum possible score of the general factor was 127. This means that the participants were highly experienced in the field of music and can be regarded as experts. The special factor "musical training" gave a mean score of *M* = 38.5 [29, 46], *SD* = 4.1 and *Max* = 49. Participants estimated their experience in music theory with *M* = 4.0 (*SD* = 1.8). The high degree of sophistication in our sample is the precondition for a high reliability of ratings.

4.2 Interestingness of tone cluster

4.2.1 Interestingness ratings

The interestingness ratings were higher for the items REF and VC1-VC3 compared to the other sounds (VC4-VC9). Overall, we observed a descending trend in perceived interestingness. Table 2 displays the mean ratings in Part 1 of the experiment.

Cluster	Density	Interes-tingness	95% CI	
	[3.5, 12.0]	*M*	*LL*	*UL*
VC1	3.5	75.5	70.1	80.9
VC2	5.0	62.7	56.3	69.1
VC3	6.0	70.7	62.6	78.8
VC4	6.5	50.3	45.5	55.1
VC5	7.0	43.6	37.4	49.8
REF	7.0	61.5	54.0	69.0
VC6	8.5	43.1	35.5	50.7
VC7	9.5	40.2	32.1	48.3
VC8	10.5	46.2	38.9	53.5
VC9	12.0	43.0	34.8	51.2

Table 2. Average ratings of interestingness (*M*) with 95 % CI (*LL* = lower limit, *UL* = upper limit).

Figure 6 displays the average ratings of interestingness with 95% confidential intervals (CI). As a fitting curve, a polynomial regression line was calculated (high strength of association R^2 = .75).

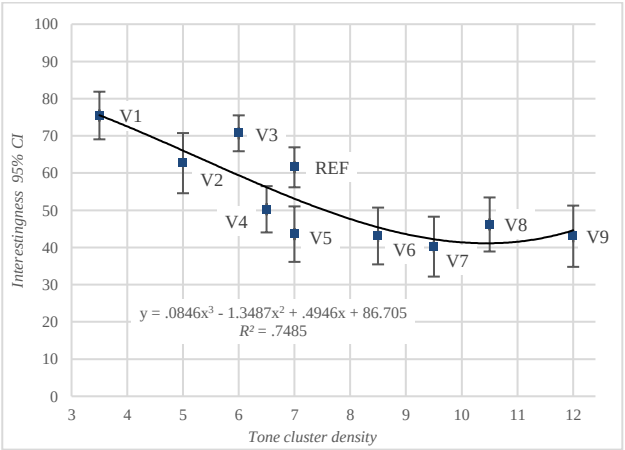


Figure 6. Average ratings of interestingness (with 95% CIs and regression line).

4.2.2 Factor analysis performed with the interestingness ratings data set

A factor analysis (varimax rotation with Kaiser normalization) was conducted to reduce the complexity of interestingness ratings. Two main components with an eigenvalue > 1 were found, explaining 67% of variance. The factor loadings are shown in Table 3. Due to their exploratory nature, the results of the factor analyses and their possible meanings will be investigated in the discussion.

Cluster	Density	Components	
	[3.5, 12.0]	1	2
REF	7.0		.757
VC1	3.5		.741
VC2	5.0		.709
VC3	6.0		.527
VC4	6.5	.834	
VC5	7.0	.837	
VC6	8.5	.852	
VC7	9.5	.881	
VC8	10.5	.914	
VC9	12.0	.910	
Eigenvalues		4.83	2.08

Table 3. Factor loadings of the interestingness ratings.[1]

4.3 Tone cluster similarity ratings

4.3.1 Recognition of the Hidden Reference

The mean overall similarity rating of the hidden reference to itself was $M = 96.4$ (95% CI [99.1, 93.7]) and was therefore very high and consistent among participants. In other words, participants rated the hidden reference when compared to the reference as nearly identical, which shows a high judgement reliability of the participants. Significant differences of means between the reference item to the other items were found. A t-test confirmed this relevant result ($t(49) = 21.82$, $p < .001$, $d = 3.09$). Only four participants rated the similarity of the hidden reference to the given reference with a value < 90. However, we decided to keep these subjects in the statistical analysis. Because of the strong recognition effect, we excluded the hidden reference in further analysis of the tone cluster density effect on the perception of similarity.

4.3.2 Similarity Ratings

Table 4 displays the similarity ratings we found for the tone clusters VC1-VC9.

Cluster	Density	Similarity Rating	95% CI	
	[3.5, 12.0]	*M*	*LL*	*UL*
VC1	3.5	25.64	19.8	31.5
VC2	5.0	35.64	28.6	42.9
VC3	6.0	43.64	37.0	50.3
VC4	6.5	44.38	37.7	51.0
VC5	7.0	56.06	49.3	62.8
VC6	8.5	54.92	48.2	61.6
VC7	9.5	45.70	39.2	52.2
VC8	10.5	47.54	41.1	54.0
VC9	12.0	50.30	43.5	57.1

Table 4. Average similarity ratings (*M*) with 95 % CI (*LL* = lower limit, *UL* = upper limit).

The distribution of the average similarity ratings might be represented as a bimodal distribution with the maxima near the density values of 7 and 12 tones per octave. The mean ratings of the items VC7-VC9 were similar. A polynomial curve fit is plotted in Figure 7 below (strength of association $R^2 = .91$).

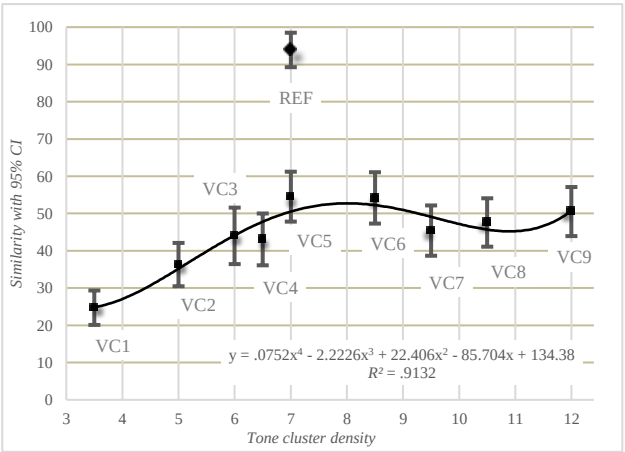


Figure 7. Average ratings of similarity (with 95% CIs and regression line).

4.3.3 Factor analysis performed on the similarity ratings

The performance of a factor analysis (varimax rotation with Kaiser normalization) with the similarity ratings revealed a 3-factor solution (see Table 5). Three main components with an eigenvalue > 1 were found, explaining 63% of the variance.

[1] All values with $r < .5$ were suspended in this table.

Cluster	Density	Components		
	[3.5, 12.0]	1	2	3
VC7	9.5	.739		
VC5	7.0	.714		
VC1	3.5	.642		
VC9	12.0	.592		
VC3	6.0		.752	
VC2	5.0		.750	
VC8	10.5		.673	
VC4	6.5			.822
VC6	8.5			.756
	Eigenvalues	2.06	1.98	1.67

Table 5. Factor loadings of the similarity ratings.[2]

4.4 Psychoacoustical measures

4.4.1 Roughness

In a last step of stimulus analysis, the psychoacoustical roughness of the 10 cluster sounds was calculated. Roughness is a psychoacoustical feature which determines the tone cluster's degree of noisiness. A timbre feature analysis is in line with the recommendations for perceptual evaluation of sounds as suggested by Fastl and Zwicker [2]. It was performed with the software *dBsonic* [10] (see Table 6).

Cluster	Density	Roughness
	[3.5, 12.0]	*M* [*centi Asper* (*cA*)]
REF	*7.0*	*17.81*
VC01	3.5	15.79
VC02	5.0	16.57
VC03	6.0	17.30
VC04	6.5	19.58
VC05	7.0	20.13
VC06	8.5	20.13
VC07	9.5	21.28
VC08	10.5	22.47
VC09	12.0	22.96

Table 6. Psychoacoustical roughness values of stimuli.

The correlation between the theoretical values of the tone cluster density and psychoacoustical roughness was r = .95. The correlation between roughness and similarity ratings was r = .74. This calculation excluded the hidden reference REF.

4.4.2 Mel-Frequency-Cepstral-Coefficient

Another feature for sound and timbre analysis as recommended by Tzanetakis & Cook [11] is the Mel-Frequency-Cepstral-Coefficient (MFCC). MFCCs have been used in speech analysis and have also been established for musical sound identification (see Loughran et al. [12]). Based on the perceptually based Mel scale, MFCCs can be used to reduce the complexity of spectral information by identification of coefficients which can be correlated with experimental data. The MFCC analysis was based on the MIR-Toolbox [13]. The MFCC0 is a feature which represents the average energy of the samples. The tone cluster density and the MFCC0 correlated by r = .98 (see Table 7). The correlation between the MFCC0 and similarity ratings (except the hidden reference) was r = .79. Tzanetakis and Cook recommend the MFCCs 2 to 6 for musical genre classification. Table 7 also displays the MFCCs 0 and 2 to 7. The correlation between the MFCCs and cluster density values of the stimuli can be seen in Table 8. This table also displays that the MFCCs correlated with the similarity ratings. The highest correlations were found between the MFCC 2, MFCC 6 and MFCC 7 and measures displayed in the columns, i. e. density, similarity and roughness.

Cluster	Density	MFCC			
		0	2	3	4
REF	*7.0*	*5.444*	*-.250*	*.992*	*.020*
VC01	3.5	4.367	-.089	1.088	.080
VC02	5.0	4.890	-.226	1.019	.040
VC03	6.0	5.439	-.164	1.033	.058
VC04	6.5	5.596	-.179	1.122	.181
VC05	7.0	5.656	-.243	1.053	.121
VC06	8.5	6.195	-.208	.990	.099
VC07	9.5	6.289	-.241	1.114	.159
VC08	10.5	6.439	-.235	1.061	.083
VC09	12.0	6.750	-.231	1.053	.106

Cluster	Density	MFCC		
		5	6	7
REF	*7.0*	*.360*	*.113*	*.147*
VC01	3.5	.359	-.110	.018
VC02	5.0	.347	.058	.035
VC03	6.0	.398	-.031	.085
VC04	6.5	.403	.115	.040
VC05	7.0	.329	.017	.093
VC06	8.5	.424	.066	.134
VC07	9.5	.374	.141	.101
VC08	10.5	.367	.105	.192
VC09	12.0	.374	.116	.172

Table 7. Values of the MFCC performance.

[2] All values with r < .59 were suspended in this table.

MFCC	Correlation coeffients *r*		
	with density	with similarity ratings (except REF)	with roughness
MFCC2	-.63	-.73	-.59
MFCC3	.02	-.29	.21
MFCC4	.29	.35	.51
MFCC5	.18	.21	.14
MFCC6	.71	.56	.68
MFCC7	.86	.68	.75

Table 8. Correlation between selected MFCCs, cluster density, and similarity ratings.

5. DISCUSSION AND CONCLUSION

The results from the factor analysis interestingness ratings may be interpreted as a confirmation of the expectation that familiarity of clusters is important in this kind of perception. The first factor can be characterized by unfamiliar cluster structures. The second factor included these chords which can be described with well-known concepts in music theory such as pentatonic or diatonic: for example, the item VC1 is a type of chord that can be found in jazz. It can be surmised that listeners are more familiar with the sound of such types of chords.

The results of the similarity rating experiment can be interpreted as a confirmation of the ability of the participants to grasp the specific quality of tone clusters. This can be observed in the high recognition rate of the hidden reference cluster in the similarity rating. Considering the theoretical background, we expected a unimodal distribution of the ratings with a maximum at the density value of the reference item. However, what we found via statistical analysis (see Figure 7) seemed to be a bimodal distribution of similarity ratings. The first peak of the regression line is positioned near the tone cluster density of 7.0 tones per octave. The second maximum is represented by the Cluster VC9 with a density of 12. This result suggests that a saturation effect in the aural perception of cluster sounds should be considered. The psychoacoustical timbre features and the theoretical tone cluster densities were highly correlated. The measure of density can be used to describe the strength of structure. This can also be seen in the correlation of psychoacoustical measures and similarity ratings. However, the factor analysis revealed further aspects to be examined in future studies.

Our findings can provide the basis for a systematic ear training for avant-garde music sounds. The insight into the density effect of tone clusters can be used not only for analysis of cluster sound music. It can also be an approach to a perceptual theory of timbre-based avant-garde music. Future research will focus on slightly noticeable differences between tone cluster structures.

Acknowledgements

The authors would like to thank A. Wolf and F. C. Thiesen for their careful reading of the paper and their helpful suggestions. We also want to thank C. Reuter for his support with the psychoacoustical timbre feature analysis.

6. REFERENCES

[1] H. D. Cowell, *New musical resources*. New York: AA Knopf, 1930.

[2] H. Fastl and E. Zwicker, *Psychoacoustics: Facts and models*. Berlin, Heidelberg: Springer, 2006.

[3] G. R. VandenBos, Ed., *APA Dictionary of Psychology*. Washington, DC: American Psychological Association, 2015.

[4] M. Kagel, "Toncluster, Anschläge, Übergänge," *Die Reihe: Informationen über serielle Musik*, vol. 5, pp. 23-37, 1959.

[5] A. Forte, *The structure of atonal music*. New Haven/London: Yale University Press, 1973.

[6] P. Lansky, G. Perle, D. Hedalm, and R. Hasegawa. Atonality [Online]. Available: http://www.oxfordmusiconline.com:80/subscriber/article/grove/music/47354

[7] N. K. Schaal, A.-K. R. Bauer, and D. Müllensiefen, "Der Gold-MSI: Replikation und Validierung eines Fragebogeninstrumentes zur Messung Musikalischer Erfahrenheit anhand einer deutschen Stichprobe," *Musicae Scientiae*, vol. 18, pp. 423-447, 2014.

[8] S. Quakenbusch, "STEP - Subjective Test and Evaluation Program," 2.00 ed. Scotch Plains, NJ: Audio Research Labs, 2004.

[9] International Telecommunation Union, "Method for the subjective assessment of intermediate quality level of audio systems BS Series Broadcasting service," ed. Geneva, 2014.

[10] dBSONIC, ".dBSONIC ", 4.501 ed. Limonest, France: 01dB-Metravib, 2012.

[11] G. Tzanetakis and P. Cook, "Musical genre classification of audio signals," *Speech and Audio Processing, IEEE transactions on*, vol. 10, pp. 293-302, 2002.

[12] R. Loughran, J. Walker, M. O'Neill, and M. O'Farrell, "The use of mel-frequency cepstral coefficients in musical instrument identification," in *International Computer Music Conference, Belfast, Northern Ireland*, 2008.

[13] O. Lartillot and P. Toiviainen, "A Matlab toolbox for musical feature extraction from audio," in *International Conference on Digital Audio Effects*, Bordeaux, 2007, pp. 237-244.

STATISTICAL GENERATION OF TWO-VOICE FLORID COUNTERPOINT

Víctor Padilla Martín-Caro[1] **Darrell Conklin**[1, 2]
[1]Department of Computer Science and Artificial Intelligence,
University of the Basque Country UPV/EHU, San Sebastián, Spain
[2]IKERBASQUE, Basque Foundation for Science, Bilbao, Spain
victor.padilla.mc@gmail.com darrell.conklin@ehu.eus

ABSTRACT

In this paper, we explore a method for statistical generation of music based on the style of Palestrina. First, we find patterns in one piece that are selected and organized according to a probabilistic distribution, using horizontal viewpoints to describe melodic properties of events. Once the template is chosen and covered with patterns, two-voice counterpoint in a florid style is generated using a first-order Markov model with constraints obtained from the template. For constructing the model, vertical slices of pitch and rhythm are compiled from a corpus of Palestrina masses. The template enforces different restrictions that filter the possible paths through the generation process. A double backtracking algorithm is implemented to handle cases where no solutions are found at some point within a generation path.

1. INTRODUCTION

In 1725, Johann Joseph Fux presented his "Gradus Ad Parnassum", a pedagogical method that breaks the learning task into well-defined graduated stages, from note against note through to florid counterpoint. This method continues to be a standard counterpoint text studied by a huge number of musicians. However, generating music based on rules is not a good stylistic approach: in fact, music from Renaissance to Romanticism can be written following basically the same rules. For example, stylistic differences between the Bach and Palestrina counterpoint is not defined by basic generic rules (parallel fifths or octaves, e.g.), and implementing specific constraints and exception to them is a very complex task. In other words, musical style has to be learnt from examples in order to model as closely as possible the composer's style. Tuning rules and exceptions can be exhausting and the rule-based system achieved becomes, in many cases, imprecise.

The option we present here is is to construct models with the real pieces of the composer. The corpus of pieces we are working with comprises 717 movements from Palestrina masses, providing a huge amount data for training statistical models of the Renaissance style.

These data are closely related with the music style without hard coding rules, but to compose a piece of music in the Renaissance style is not so simple. Counterpoint is full of imitations, canons, motifs, augmentations, etc. and such devices are not easily captured by a first-order Markov model [14]. For solving these limitations and to provide coherence to the generated pieces, we take a piece from the corpus and obtain the main patterns based on different viewpoints, as will be outlined in Section 3. The patterns are used to cover the template piece. In generated music the rhythm of the template piece is retained.

2. PRIOR WORK

The generation of new music based on rules, or expert systems, has a long tradition, from the works in 1955 of Hiller and Isaacson [1] using the ILLIAC computer at the University of Illinois. More recent work is the article of Ebcioglu implementing rules for counterpoint and Bach chorales [2]. Herremans and Sorensen work with different counterpoint species using a variable neighborhood search algorithm [3]. Komosinski and Szachewicz [4] address the difficulty of evaluating penalty (or reward) values for each broken (or satisfied) rule. A simple additive rule weighting function is weak and they propose to use the dominance relation. Implementing rules by fuzzy logic to generate two-voice first species counterpoint is analyzed by Yilmaz and Telatar [5].

On the other hand, research that uses machine learning technologies has been increasing in recent years. One possible way is to work with neural networks [6] that learn from the context developing a Bach-style choral harmonization. The probabilistic approach (HMM) has been studied by Allan and Williams using a Bach chorale corpus [7]. Focused on counterpoint generation and HMM, Farbood and Schoner [8] work with first-species counterpoint. The method uses Markov chains which capture the rules of counterpoint using probabilistic tables for harmony, melody, parallel motion, and cadences.

David Cope's Experiments in Musical Intelligence (EMI) is a system of algorithmic composition created and developed since 1981. His approach is based on Transition Networks analogous to the historical model of the musical dice game, but detecting and deciding the components autonomously. The semantic classification of the material is carried out by a system called SPEAC [9] (statement, preparation, extensions, antecedent and con-

sequent). The SPEAC system is inspired by the analysis method developed by Heinrich Schenker.

Regarding the detection of melodic phrases in the masses of Palestrina, Knopke and Jurgensen [10] describe a system based on the use of suffix arrays to find repeated patterns. The main drawback of their system is the requirement for exact matching of the patterns. Augmentations, diminutions or non-exact intervals (fourth by fifth or third minor by major) are not considered.

Many sequential pattern mining methods have been developed in the last decade, such as SPADE, PrefixSpan, GSP, CloSpan or BIDE [11]. In our research, for analysing patterns, we are using a gap-BIDE [12] algorithm with zero gaps between sequences as will be explained in 4.2.

For slicing and obtaining patterns from scores, we are using the concept of horizontal and vertical viewpoints. This idea of viewpoints was developed and refined by Conklin [13,14,15,16] and will be described in Section 4.

This paper is organized as follows. Section 3 justifies the corpus chosen. Section 4 develops the idea of horizontal and vertical viewpoints and the algorithm to find patterns. That section also explores the concept of probability with respect to zero- and first-order Markov models of the Palestrina corpus. Section 5 explains the restrictions imposed by the template and the results obtained.

3. PALESTRINA'S MASSES

The corpus of pieces we are working with consists of 101 masses composed by Palestrina. These masses were published between 1554 and 1601, after his death in 1594. The date of composition of the different pieces is very difficult to determine, and each mass consists of various movements: Kyrie, Gloria, Credo, Sanctus, Benedictus, Agnus Dei. Each movement is divided into sections based on the text. The masses and the movements vary in number of voices from three to six. For example, Benedictus in many masses is written in three voices and Kyrie in five or six. Below is the corpus of pieces we have, using the data of music21[1], which is a Python-based toolkit for computer-aided musicology developed by MIT.

Mass part	Pieces
Agnus	186
Benedictus	99
Credo	98
Gloria	101
Kyrie	129
Sanctus	104
Total:	**717**

[1]http://web.mit.edu/music21/

There are several arguments for using the masses of Palestrina as a test collection for our system. They are a model for a standard Renaissance style in counterpoint. Many universities and conservatories teach this style as basic training for new students in composition. Another important aspect is the homogeneity of the corpus of pieces. There are not significant differences in style between the first and the last mass, and the number of pieces is big enough to build a probabilistic model. Regarding this point and taking into account just two voices, the number of vertical slices available is almost 350,000 which provide enough information for constructing a reasonably accurate first-order Markov model, as is explained in the next section.

4. VIEWPOINTS FOR PATTERN DISCOVERY

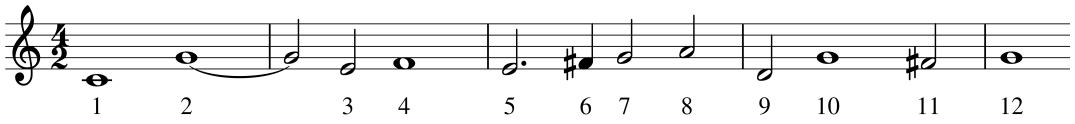
For generation of polyphony, both horizontal (melodic) and vertical (harmonic) aspects must be modelled. In this work, we implement the concept of *linked viewpoints*, from the horizontal and vertical perspective. As an overview [15], a viewpoint system is a collection of independent views of the musical surface each of which models a specific type of musical phenomena. A piece of music is therefore transformed into a higher level description derived from the basic surface representation. For every viewpoint a *viewpoint sequence* function transforms a sequence of basic events into a sequence of defined viewpoint elements. A linked viewpoint is a combination of two or more viewpoints that models other viewpoints simultaneously.

4.1 Horizontal viewpoints

Each phrase of Palestrina music is treated as a sequence of linked viewpoints. To better understand the concept of viewpoint, we take a melody of Palestrina. The sequence of notes is converted to a sequence of features derived from the musical surface (Figure 1), for example, absolute pitch, name of note (spell), melodic contour, duration contour, interval or a group of interval joined (scalestep). A pattern is a sequence of features $(v_1, \dots, v_l)$ where each v_i is a feature (e.g. scale step linked with contour of duration, as specified in Equation 1).

The scalestep viewpoint groups successive intervals and is flexible enough to find patterns in Renaissance style. The values of that viewpoint are:

- Unison and Octave (J18)
- Minor second and Major second (Mm2)
- Minor third and Major third (Mm3)
- Perfect fourth and Perfect fifth (J45)
- Minor sixth – Major sixth (Mm6)
- Minor seventh - Major seventh (Mm7)



	1	2	3	4	5	6	7	8	9	10	11	12
pitch	60	67	64	65	64	66	67	69	62	67	66	67
spell	C	G	E	F	E	F#	G	A	D	G	F#	G
contour(pitch)	⊥	+	-	+	-	+	+	+	-	+	-	+
contour(dur)	⊥	+	-	+	-	-	+	=	=	+	-	+
diaintc	⊥	J5	m3	m2	m2	M2	m2	M2	J5	J4	m2	m2
scalestep	⊥	J45	Mm3	Mm2	Mm2	Mm2	Mm2	Mm2	J45	J45	Mm2	Mm2

Figure 1. Different viewpoints from a melody of Palestrina.

The repetitions of patterns in Palestrina are not merely exact transpositions of intervals. For example, a minor second can be converted to a major second, as is shown in Figure 2. Using this scalestep viewpoint, equivalent intervals, from the point of view of imitation, are grouped into the same category.

4.2 Finding patterns

Each file from the corpus is converted to a viewpoint sequence, separating phrases by rests. A link between-viewpoints is defined using the constructor ⊗. The linked viewpoint for discovering patterns is:

$$\text{scalestep} \otimes \text{contour(duration)} \quad (1)$$

This particular linking of viewpoints allows the discovery of augmented and diminished patterns and melodic inversions.

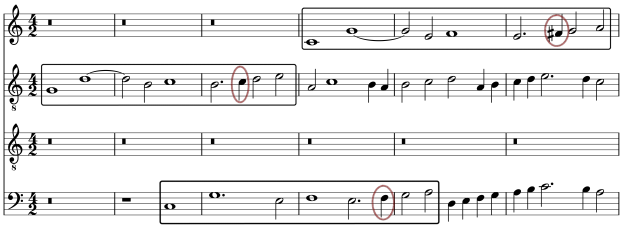


Figure 2. Agnus from *Beata Marie Virginis*. Palestrina. Patterns with different intervals.

Data mining [11] is the computational process of discovering patterns in a large data set. This interdisciplinary subfield of computer science is growing, and the number of algorithms and researchers in the field highlights its importance. Sequential pattern mining has become an essential data mining task, with broad applications, including market and customer analysis, web log analysis, pattern discovery in protein sequences, etc. A sequence of musical features can be analysed as a sequence of DNA and, taking ideas from biology, to identify similar and repeated patterns through the string of elements.

Well known algorithms for sequential pattern mining [10] are SPADE (Sequential PAttern Discovery using Equivalence classes), PrefixSpan (Prefix-projected Sequential pattern mining), GSP (Generalized Sequential Pattern algorithm), CloSpan (Closed Sequential pattern mining) or BIDE (BI-Directional Extension). In our experiments we are using gap-BIDE, an extension of the BIDE algorithm for mining closed sequential patterns with possible gap constraints. Currently, we are working at zero gap level without taking into account gaps in the sequences.

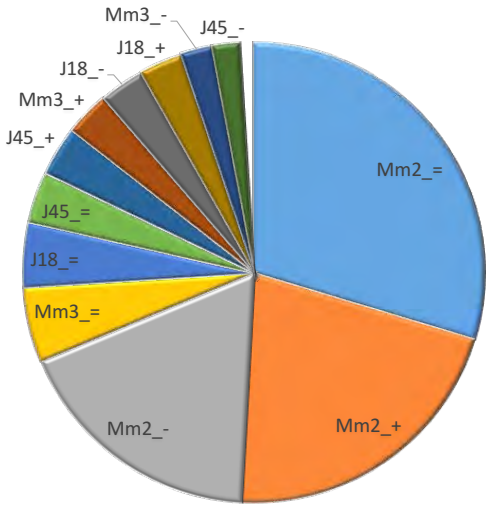


Figure 3. Distribution of different features in the corpus using Equation (1)

4.3 Ranking patterns

We establish a ranking of patterns based on a binomial distribution that computes the probability of obtaining an observed number of occurrences in a given number of sequence positions within the template piece.

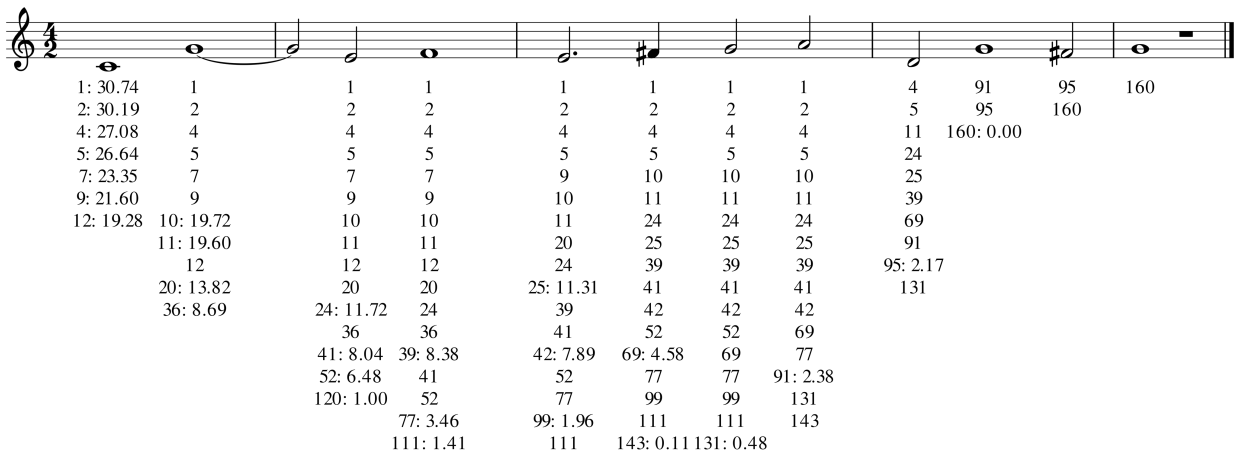


Figure 4. Agnus from *Beata Marie Virginis* of Palestrina (Agnus.krn). Palestrina. Different patterns ordered by binomial distribution. The first number gives the rank of the pattern, and the second the $\mathbb{I}$ value for the first position of a pattern.

Figure 3 shows a distribution of the values of the viewpoint defined in (1). For example, Mm2_= indicates a scale step of a minor or major second with an equal duration, or J45_+ indicates a scale step of perfect fourth or fifth where the second note has a higher duration. Clearly, the most probable interval is the second (69.4%) divided into same duration (30.0%), higher duration the second note (21.4%) and lower duration the second note (18.0%). The rest of the intervals have a much lower probability. This illustrates that, for example, patterns comprising predominantly Mm2_= features are not surprising and will not be significant unless they occur very frequently in the piece. The binomial pattern ranking, as described below, handles these effects in the piece.

One possible way of coding the patterns is as follows (Table 1). The odd lines (grey) indicate the array of linked features and the even ones (white) the position of the pattern in different phrases. The three numbers specifying the position are the indices of the phrase, start note and end note of the pattern.

Patterns
<J45_+> <Mm3_-> <Mm2_+> <Mm2_-> <Mm2_-> <Mm2_+> <Mm2_=>
(4, 0, 7), (13, 0, 7), (27, 0, 7)
<J45_+> <Mm3_-> <Mm2_+>
(0, 0, 3), (4, 0, 3), (13, 0, 3), (21, 0, 3), (27, 0, 3)
<J45_+> <Mm3_-> <Mm2_+> <Mm2_+> <Mm2_-> <Mm2_+>
(0, 0, 6), (21, 0, 6)

Table 1. Example of coding patterns. Agnus from *Beata Marie Virginis* of Palestrina.

The background probability (b_p) of a pattern $p = (v_1, \dots, v_l)$ must be estimated, for example using a zero-order model of the corpus

$$b_p = \prod_{i=1}^{l} c(v_i)/N \qquad (2)$$

where:

- $c(v_i)$ is the total count of feature v_i,
- N is the total number of places in the corpus where the viewpoint is defined.

Using the background probability of a pattern, its interest $\mathbb{I}$ can be defined using the binomial distribution, which gives the probability of finding at least the observed number of occurrences of the pattern.

$$\mathbb{I}(p) = -\ln \mathbb{B}_{\geq}(k_p; t_p, b_p) \qquad (3)$$

where:

- $\mathbb{B}_{\geq}$ gives the cumulative probability (right tail) of the binomial distribution,
- t_p approximates the maximum number of positions that can be possibly matched by the pattern,
- k_p is the number of times the pattern appears in the template piece.

Figure 4 is one example of different patterns found in one fragment of Agnus from *Beata Marie Virginis* of Palestrina, ordered by (3). The number followed by a colon (:) indicates the $\mathbb{I}$ for each pattern. This template will be used for creating the new piece as is explained in the next section.

4.4 Vertical viewpoints. Markov model

For constructing the Markov model, two voices are selected and cut into slices (see Figure 5). In this first approach, we have taken the highest and lowest voice for a better result, removing the intermediate. Usually, the music that follows harmonic constraints entrust to the lower part (bass) an important role in the harmonic context, while the higher (soprano) is more appropriate for defining melodies.

The slicing process is the same as the method explained by Conklin [17], dividing when a new event appears in one voice. In our example, for simplicity, we do not retain continuations or ties (the full expansion method of [17]). In Renaissance vocal music, the repetitions or ties sometimes depend on the text and, in the new score, the durations are going to be obtained from the template.

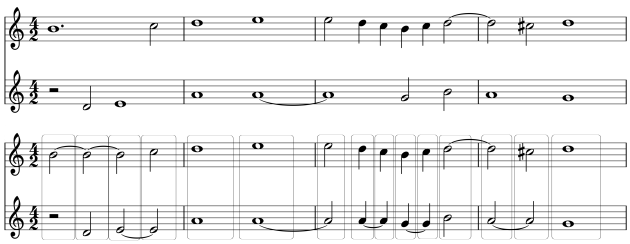


Figure 5. Example of two-voice slicing.

Taking into account pitch and duration, the number of slices is 347,748. The zero-order Markov model is calculated counting the number of repeated slices and dividing by the total. The number of different slices is 1582 distributed as is shown in Figure 6. The vertical axis is the number of repetitions (logarithmic scale) and the horizontal the slice ordered by repetitions. Counting the number of unique next slices (first-order Markov model), ordered by the zero-order model, the results are shown in Figure 7, where the number of different paths ranges from 0 and 183.

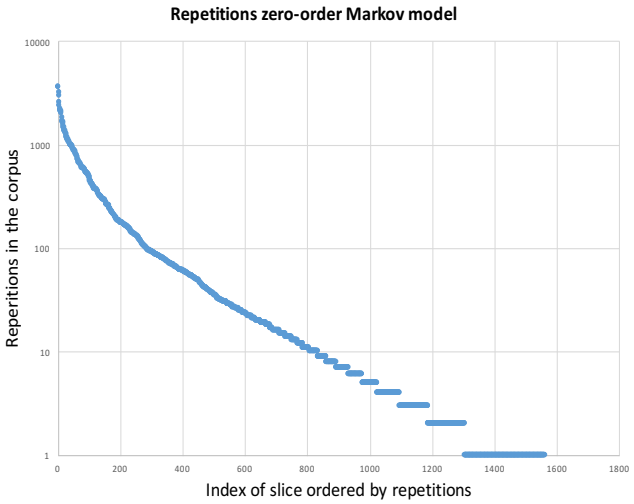


Figure 6. Zero order distribution of repetitions.

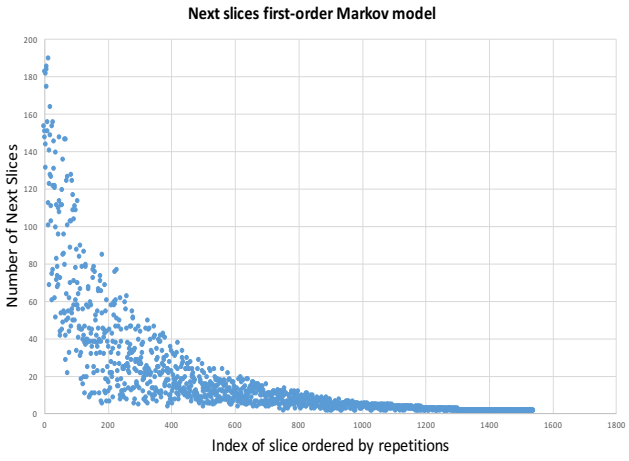


Figure 7. Distribution of unique next slices, first-order Markov model.

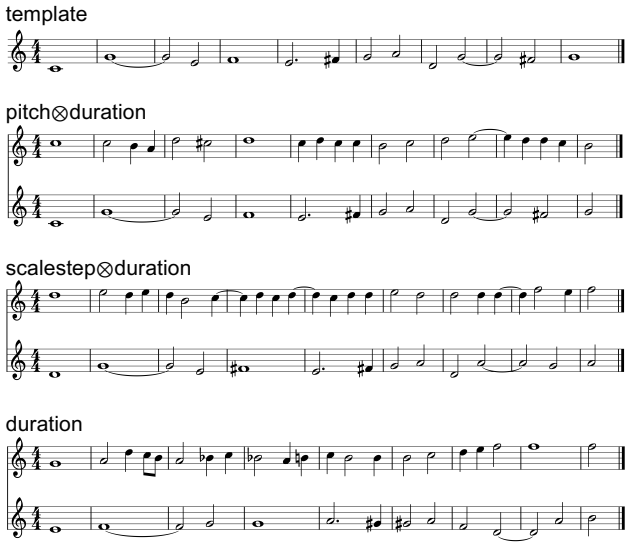


Figure 8. Generation of upper voice based on different constraints in lower voice.

The piece now can be treated as a sequence of regular simultaneities where it is possible to apply different constraints that filter the possible paths. For example, based on this melody as a pattern (Figure 8), we illustrate the system with different restriction levels for creating a new upper voice. The upper voice is generated applying a random walk among the possible vertical slices using a first-order model. It is a short phrase and it was easy to find solutions through forward generation with just one template and different viewpoint constraints in the lower voice. Ranking from strongest to weakest, and using linked viewpoints, they are labelled as pitch⊗duration, scalestep⊗duration and duration.

5. APPLYING THE MODEL TO THE TEMPLATE

This section explains a method for generating new music based on a piece from the corpus. The idea is to take the patterns of this piece, which guarantees coherence, and fill the template with the slices and probabilistic paths obtained by the first-order model. The steps are as described below.

5.1 Forward generation

For generating new music, one piece from the corpus is chosen and patterns are discovered in the piece using the viewpoint scalestep⊗contour(duration) as mentioned earlier. Once we have all the possible patterns, a greedy algorithm is used for covering the score. This algorithm takes the better patterns from left to right. For practical purposes, in the case of simultaneous patterns in two voices, they are evaluated removing the item that has a lower score. The template is, therefore, divided into different regions that are separated based on the patterns. Figure 9 shows the first six measures of the Agnus from

Beata Marie Virginis and how the patterns are found in the different voices. For simplicity, we will take the upper and lower voice for the generation.

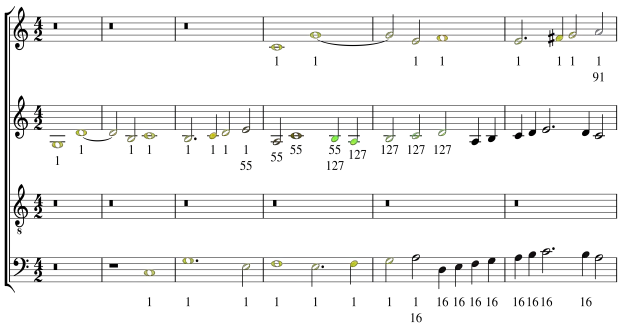


Figure 9. Agnus from *Beata Marie Virginis* of Palestrina. Covering the piece using a greedy algorithm.

Once the final template is decided, for constraining notes within areas covered by patterns, the viewpoint

$$\text{scalestep} \otimes \text{duration} \otimes \text{contour(pitch)} \quad (4)$$

is used. Note that this represents a slightly more restrictive pattern than that used for pattern discovery (1), in that the regions are also required to conserve pitch contour. The generated music therefore conserves the abstract qualities of scale step, duration contour, and pitch contour. Further, in this paper the exact durations from the Palestrina template are used, therefore the conservation of duration is assured.

Regarding the corpus of pieces, some of them are composed of just three voices (e.g. Benedictus). For testing purposes, we take Benedictus from the mass *Descendit angelus Domini* as a template and proceed with the next steps:

- First, remove internal voices retaining the highest and lowest.
- Divide the template into regions organized by the patterns. If the region is a pattern, the viewpoint shown in Equation (4) is used for horizontal restrictions. If the region is not a pattern, just the duration viewpoint remains.
- The vertical slices are filtered by the different constraints. If at one point it is not possible to find next slice, a backtracking algorithm is performed (see Section 5.2).

There is a probability associated with each new slice in the first-order Markov model. Different results will be obtained choosing a different range of probabilities, as will be commented in Section 6.

5.2 Double backtracking algorithm

Due to the severe restrictions forced by the template, it is possible to encounter some points where all slices to continue the piece have zero probability at the slices generated. This problem was due to the bottleneck arising from the availability of very few possibilities for some slices of the corpus. To solve this problem, a double backtracking algorithm has been implemented at two different levels, pattern and template. At the pattern level, the system goes one, or several steps back if no possible solutions are obtained for some slice. If the backtracking at pattern level reaches the first slice, the system goes back one (or several steps back) from the patterns of the template. This method is faster and permits a scattered group of solutions uniformly distributed.

Figure 10 shows one piece generated by this method. The two upper systems are the original Benedictus from *Descendit angelus Domini*. The colors indicate the patterns found by the greedy algorithm. The three lower systems are the music generated based on the upper and lower voice from the template.

6. CONCLUSIONS AND NEXT STEPS

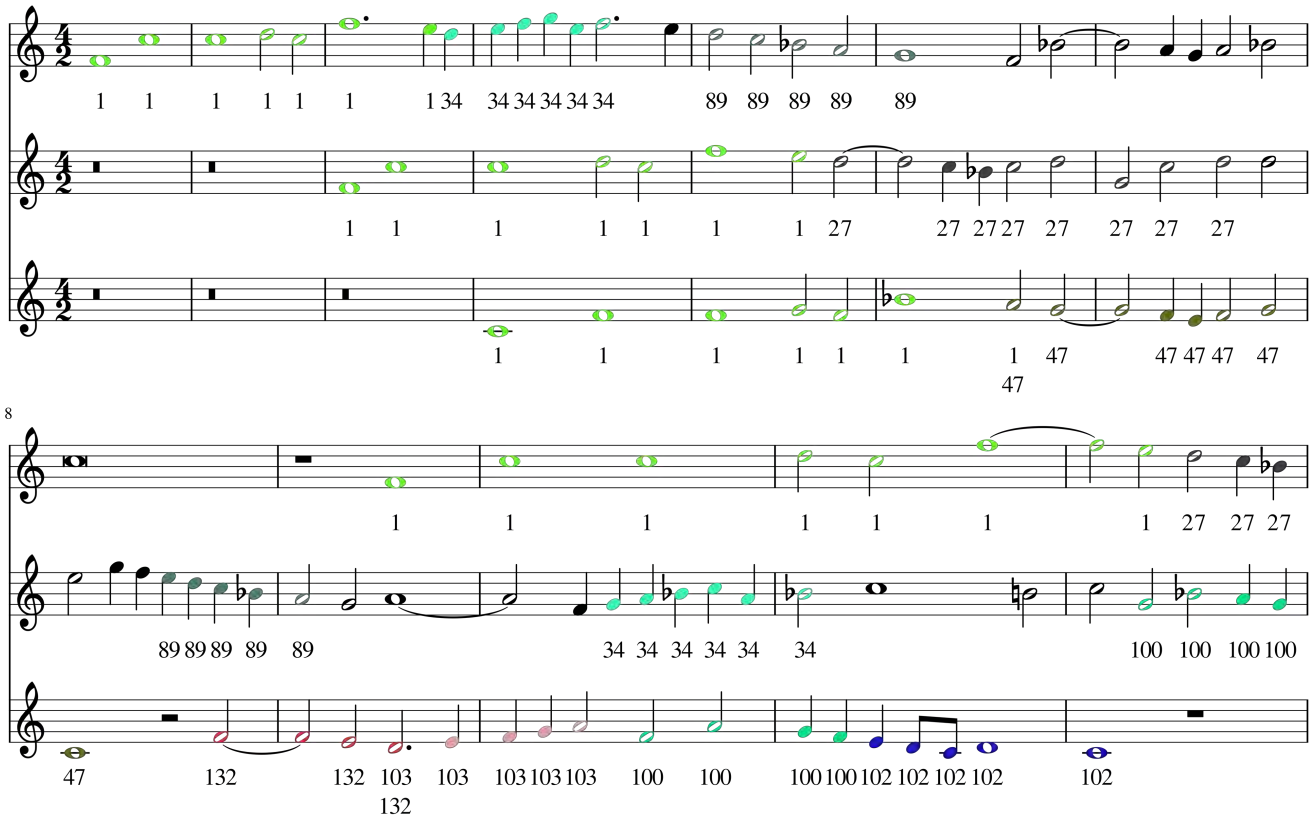
This paper presents a method for generating new music based on the corpus of masses of Palestrina. The system, for practical purposes, is limited to two voices taking a template from the corpus without overlapping patterns.

The gap-BIDE algorithm and the binomial distribution explained in Section 4.2 works correctly in most of the cases, and the ranking of the patterns discovered is related to the importance of the pattern in the piece. The greedy covering algorithm is quite simple and will be revised in a future version. Though this aspect is not the main goal of the project, a deeper research finding strengths and weaknesses of the template extracted should be done.

Regarding the Markov model, the zero- and first-order are a good approach for ensuring correctly linked slices with rhythm and pitch constraints. The first-order Markov model prevents "weak" linked chords with grammatical errors such as parallel fifths, parallel octaves, without implementing specific rules. This model does not organize harmonic regions, and "non-idiomatic" melodic movements can appear. In this sense, a second-order model implementation could be an improvement for generating better melodies, but the training data would decrease exponentially. The main goal of this work is that the template complements some weaker aspects of the first-order Markov model and provides some kind of melodic coherence. In other systems, (David Cope's EMI, e.g.), the coherence is achieved analyzing bigger slices of the pieces. In our case, the slices are reduced to the minimum rhythmic value and the possible structural information obtained, sparse. The template, therefore, provides the necessary scaffolding for the melodic ideas.

Section 5.2 commented on the double backtracking algorithm performed if no solution is found. The processing time is very high to find solutions using random walks when the group of optimum linked slices is very small, and in some cases there may not be a solution due to the hard requirements of the patterns selected. The

Benedictus from *Descendit angelus Domini*. Template from the original score



New music generated based on the upper and lower voice from the previous template.

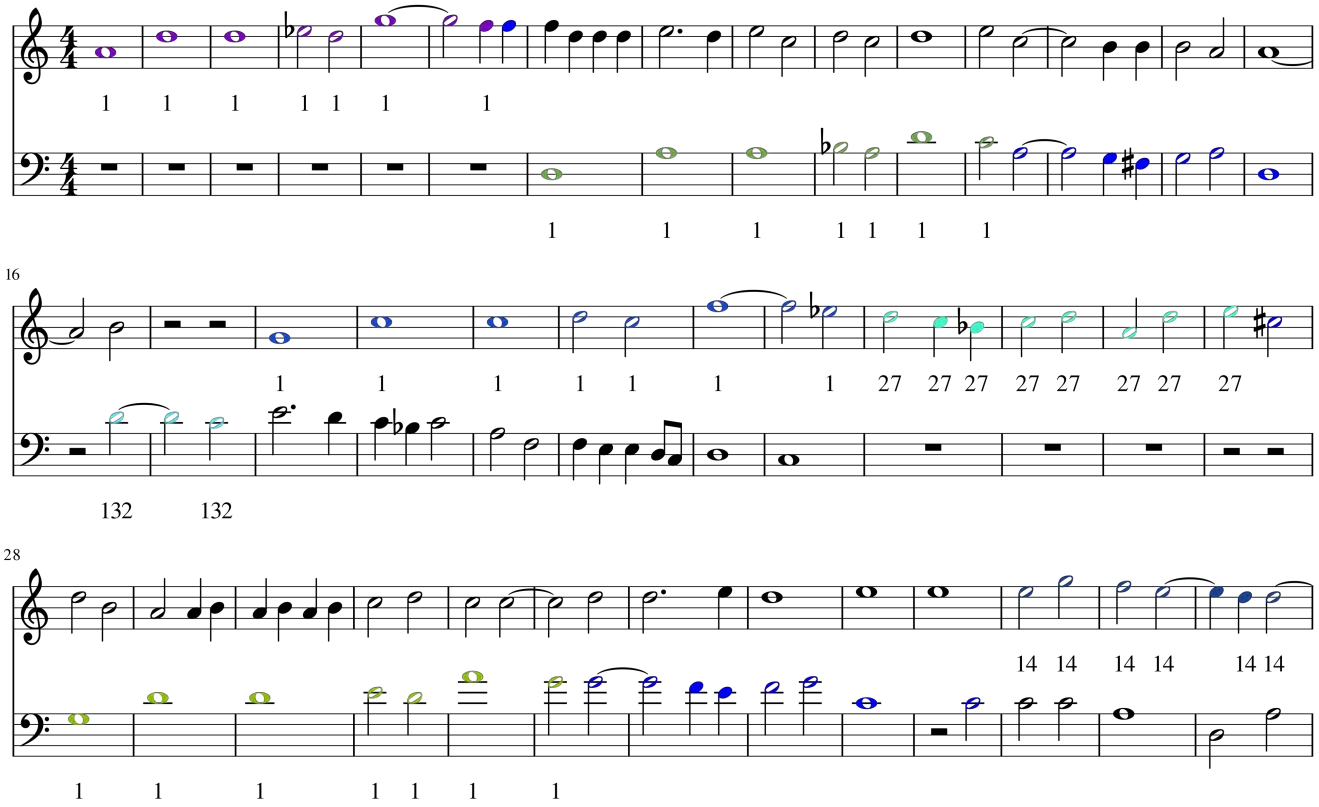


Figure 10. Template and new music generated. Benedictus from *Descendit angelus Domini* of Palestrina (Benedictus_19.krn). The upper score is the original from Palestrina. The lower score is two voices generated based on higher and lower part. The colors identify different patterns.

backtracking algorithm is faster than a simple random walk and provides a group of solutions homogeneously distributed. Another possibility that could be implemented in a future version is a depth-first search to explore all the different paths, which might lead to more heterogeneity in the results.

This model is made and tested for two voices, but it is possible to extend to three or more voices using different viewpoints such as vertical intervals and duration. The zero-order Markov model will grow significantly, and the slices with higher probabilities will possibly decrease, augmenting the dead-end solutions, but hopefully, the corpus is large enough to find paths and create new and interesting pieces.

Acknowledgments
This research is supported by the project Lrn2Cre8 which is funded by the Future and Emerging Technologies (FET) programme within the Seventh Framework Programme for Research of the European Commission, under FET grant number 610859.

7. REFERENCES

[1] L. Hiller and L. Isaacson, "Musical composition with a high-speed digital computer," *Machines Models of Music*, pp. 9-21, 1958, Reprint of original article in Journal of Audio Engineering Society.

[2] K. Ebcioğlu, "An Expert System for Harmonizing Four-part Chorales," *Computer Music Journal*, vol. 12, no. 3, pp. 43-51, 1988.

[3] D. Herremans and K. Sörensen, "Composing fifth species counterpoint music with a variable neighborhood search algorithm," *Expert Systems with Applications*, vol. 40, no. 16, pp. 6427-6437, 2013.

[4] M. Komosinski and P. Szachewicz, "Automatic species counterpoint composition by means of the dominance relation," *Journal of Mathematics and Music*, vol. 9, no. 1, pp. 75-94, 2015.

[5] A. Yilmaz and Z. Telatar, "Note-against-note two-voice counterpoint by means of fuzzy logic," *Knowledge-Based Systems,* vol. 23, no. 3, pp. 256-266, 2010.

[6] H. Hild, J. Feulner, and W. Menzel, "HARMONET: A neural net for harmonizing chorales in the style of JS Bach," *Advances in Neural Information Processing*, pp. 267–274, 1992.

[7] M. Allan and Williams C., "Harmonising chorales by probabilistic inferences," *Advances in Neural Information Processing Systems*, vol. 17, pp. 25-32, 2004.

[8] M. Farbood, and B. Schoner, "Analysis and synthesis of Palestrina-style counterpoint using Markov chains," in *Proceedings of the International Computer Music Conference*, Cuba, 2001, pp. 471-474.

[9] D. Cope, "Virtual music: computer synthesis of musical style," MIT Press, Cambridge, 2001.

[10] I. Knopke and F. Jürgensen, "A system for identifying common melodic phrases in the masses of Palestrina," *Journal of New Music Research*, vol. 38, no. 2, pp. 171-181, 2009.

[11] I. Khan and A. Jain, "A Comprehensive Survey on Sequential Pattern Mining," *International Journal of Engineering Research and Technology*, vol. 1, no. 4, June 2012.

[12] Li Chun and Wang Jianyong, "Efficiently Mining Closed Subsequences with Gap Constraints," in *SIAM*, 2008, pp. 313-322.

[13] D. Conklin, "Discovery of distinctive patterns in music," *Intelligent Data Analysis,* vol. 14, no. 5, pp. 547-554, 2010.

[14] D. Conklin, "Music generation from statistical models," in *Proceedings of the AISB 2003 Symposium on Artificial Intelligence and Creativity in the Arts and Sciences*, Aberystwyth, Wales, 2003, pp. 30-35.

[15] D. Conklin and I. H. Witten., "Multiple viewpoint systems for music prediction," *Journal of New Music Research*, vol. 1, no. 24, pp. 51-73, 1995.

[16] D. Conklin and C. Anagnostopoulou, "Representation and Discovery of Multiple Viewpoint Patterns," in *Proceedings of the International Computer Music Conference*, San Francisco, 2001, pp. 479-485.

[17] D. Conklin, "Representation and discovery of vertical patterns in music," in *Music and Artificial Intelligence*, C. Anagnostopoulou, M. Ferrand, and A. Smaill, Eds.: Springer-Verlag Lecture Notes in Artificial Intelligence 2445, 2002, pp. 32-42.

[18] M. Bergeron and D. Conklin, "Temporal patterns in polyphony," in *Mathematics and Computation in Music*. Berlin, Heidelberg: Springer, 2009, pp. 32-42.

[19] M. Bergeron and D. Conklin, "Subsumption of vertical viewpoint patterns," in *Mathematics and Computation in Music*, MCM 2011 Third International Conference, Ed. Paris, France, Springer, 2011, pp. 1-12.

[20] D. Conklin and M. Bergeron, "Feature set patterns in music," *Computer Music Journal*, vol. 32, no. 1, pp. 60-70, 2008.

INTERACTION WITH A LARGE SIZED AUGMENTED STRING INSTRUMENT INTENDED FOR A PUBLIC SETTING

Jimmie Paloranta, Anders Lundström, Ludvig Elblaus, Roberto Bresin, Emma Frid
KTH Royal Institute of Technology
{jimmiep, andelund, elblaus, roberto, emmafrid}@kth.se

ABSTRACT

In this paper we present a study of the interaction with a large sized string instrument intended for a large installation in a museum, with focus on encouraging creativity, learning, and providing engaging user experiences. In the study, nine participants were video recorded while interacting with the string on their own, followed by an interview focusing on their experiences, creativity, and the functionality of the string. In line with previous research, our results highlight the importance of designing for different levels of engagement (exploration, experimentation, challenge). However, results additionally show that these levels need to consider the users age and musical background as these profoundly affect the way the user plays with and experiences the string.

1. INTRODUCTION

When designing interactive installations in public settings such as museums and art galleries, designers face new challenges, considering the wide variety of possible users, the impact of the surrounding environment and the durability and reliability necessary for long-term (and sometimes unexpected) user interaction.

In this paper, we present a study of a large sized augmented string instrument, with focus on how the participants approach the string and how it can encourage creativity and provide an engaging user experience. The instrument, as presented here, acts as a formative prototype for a future museum installation at the new Scenkonstmuseet that opens in 2017, in Stockholm, Sweden. The final installation will be called LjudSkogen/Sound Forest and consist of 5 similar strings in a dedicated room. The string metaphor was chosen for its affordances and familiarity, with the aim of making it as intuitive as possible [1] so that anyone, regardless of musical background should be able to play and be creative with the instrument.

2. BACKGROUND AND THEORY

2.1 Interactive Installations in Public Settings

Several comprehensive studies have looked at interactive exhibits from the perspectives of engagement and learning [1–5]. A general conclusion in these studies is the importance of rewarding both initial and prolonged engagements with an installation [1], where the experience of initial interaction is crucial as it often determines whether the visitor will continue to interact with the installation or not [5]. After this initial phase, the complexity should be increased for prolonged engagement with the exhibit [2]. This can for example be achieved by offering new opportunities and challenges for exploration and experimentation. Otherwise, visitors tend to leave the exhibit once they have figured out how the system works [3].

While kids and families form the main part of museum visitors [6], studies have highlighted the challenges of designing for the broad target group that installations in public settings typically aim for. For instance, Taxen et al. [7] noted that people have very different approaches when interacting with installations and highlighted the importance of accounting for these approaches when designing exhibits. Due to this irreducible complexity [2], iteration and evaluation with people throughout the development process is key when designing for these settings [8], as the full complexity of an exhibits interactive features can be seen only through the eyes of the visitors, no matter how experienced the designers are. A study by Campos et al. [4] also mentions the challenge that arises when an interactive installation is finally deployed, as many aspects are impossible to model or test by means of early prototypes (like the surrounding environment's impact on the experience).

2.2 Interactive Augmented Strings

Within the field of augmented instruments, much work has been done on string instruments like guitars, violins and pianos [9–16] with particular interest in sensing gestures or position of a player's fingers on the instrument. Most previous approaches to this have been based on image analysis, as noted by Guauas et al. [12] who in their study instead proposed a method of capacitive sensing. While this method successfully managed to capture gestures and touch, as shown in similar studies by McPherson et al. [13] and Tobise and Takegawa [16], it has been less successful for position measuring due to body impedance causing too

much interference with the system [9]. In another study by McPherson et al. [14], optic sensors were used to accurately measure the height position of a pressed key on a piano, while Newton and Marshall [15] used infrared sensors to detect strumming motions on an augmented guitar.

2.3 User Experience

In his book Art as Experience [17], pragmatic philosopher John Dewey defined experience as constant and something that occurs continually, as we are always in the process of living. Dewey discusses the definition of an "aesthetic experience" and experiences in the world of the arts, arguing that every prosaic experience can be of aesthetic quality, since all experiences can be rich and fulfilling.

Building on Deweys pragmatic approach to experience, Wright and McCarthy provide a framework for analysing experiences in their book Technology As Experience [18]. The framework consists of four intertwined "threads of experience" and is accompanied by six non-linear sense-making processes. While the four threads outline the compositional, emotional, sensual and spatio-temporal elements of an experience, the sense-making processes (anticipating, connecting, interpreting, reflecting, appropriating, recounting) are of more interest to this study as they dwell deeper into the personal traits of the user during an experience and are therefore more evaluable from a user experience perspective.

- Anticipating: refers to the expectations, possibilities and ways of making sense that we bring prior to the event of the experience.
- Connecting: refers to the immediate, pre-conceptual and pre-linguistic sense or feeling of a situation encountered.
- Interpreting: refers to the discerning of the narrative structure and possibilities of the unfolding experience, what has happened and what is likely to happen.
- Reflecting: refers to the judgments made about the experience as it unfolds, which happens at the same time as interpreting.
- Appropriating: refers to relating the experience to our own sense of self, in context to our personal history and future.
- Recounting: refers to telling the experience to others or ourselves, which gives us the opportunity to find new possibilities and meanings in it.

2.4 Creativity

Creativity is a big part of experience, both in the views of philosophers and researchers. Apart from Dewey, pragmatic philosopher Mikhail Bakhtin also inspired Wright and McCarthys work in Technology as Experience. Bakhtin believed "that to live is to create", and that the act we describe as creative is just extensions of the sorts of activity we perform all the time [19], which can be reflected in Wright and McCarthys views that "in an open world, all action is creative, a fresh use of intelligence producing something surprising and new every time" [13].

According to psychologist Robert Sternberg, most investigators within the scientific field would agree on the general definition of creativity as "the process of producing something that is both original and worthwhile" [20], but what is "worthwhile" is a highly subjective notion and therefore also complicates evaluation. Within psychology however, divergent thinking (exploring many possible solutions to a set problem) is often seen as correlated with creativity. For instance, educational psychologist Frank E Williams [21] has used it as a measure of creativity. Drawing from the foundations of divergent thinking, Williams created a model of eight different creative skills that were used to learn and measure creativity among students, called Williams Taxonomy [21]. The skills were *fluency* (the ability to generate many ideas so that there is an increase of possible solutions), *flexibility* (the ability to produce different categories of ideas), *elaboration* (the ability to add on an idea), *originality* (the ability to create unique ideas), *complexity* (the ability to conceptualize multifaceted ideas), *risk-taking* (the willingness to be daring and try new things), *imagination* (the ability to dream up new ideas) and *curiosity* (the trait of exhibiting probing behaviours, asking, searching and wanting to know more about something).

3. METHOD

To investigate how presumptive visitors might interact with and perceive a large string augmented instrument we developed a first interactive prototype and let nine participants (see Table 1) with different background interact with the prototype on their own. The participants were not given any specific instructions regarding how to interact with the prototype. The participants represented different groups of the museums envisioned target audience: children, parents and young adults with musical interest. Three children in the ages of 9-11 (C1-C3), four young adults in the ages of 23-29 (Y4-Y7) and two parents, both 53 years old (P8-P9), participated in the experiment. The children were all male, while half of the young adults and parents were female and male, respectively.

The procedure of the user tests was as follows: first, a brief interview was held to gather information about the participants experience of music and museums. Then, the participant was left alone with the string in a lab room. No prior explanation of how the string would react to interaction was given, participants were only told that the string would be a part of a music installation at Scenkonstmuseet. The participants were told that they were free to play around and explore the string for as long as they wanted. To increase the probability of capturing their thoughts and considerations in action they were encouraged to think aloud during the interaction with the string. Lastly, semi-structured interviews were conducted in three parts. The first part dealt with the different processes of the users experience (anticipating, connecting, interpreting, reflecting, appropriating and recounting), based on the framework provided by Wright and McCarthy [18]. The second

User	*Age*	*Gender*	*Musical Background*
C1	9	Male	No previous experience
C2	10	Male	Guitar, one semester
C3	11	Male	No previous experience
Y4	23	Male	Drums, 4 years when younger
Y5	24	Female	Piano, 4 years when younger
Y6	27	Male	Piano, 9 years
Y7	28	Female	Piano and violin, 21 years
P8	53	Male	No previous experience
P9	53	Female	Piano and Guitar, 4 years when younger

Table 1. Participants in the user test.

part was based on Williams Taxonomy [21] and dealt with the creative skills displayed by the users (fluency, flexibility, elaboration, originality, complexity, risk-taking, imagination and curiosity) during interaction. The last part focused on the functionality of the prototype and the string's material. The questions were first written for adults and then reformulated using simpler vocabulary in order to be more suitable for the children (for example, the question Did you feel like you could create something original? was changed to Did you feel like you could create something new?).

The participants interaction with the string and the interviews were video recorded. The interviews were then thematically analysed for common, reoccurring themes. These themes where then used for further video analysis of what actually seemed to occur during the interaction, with focus on the processes of the users experiences and the creative skills displayed.

4. THE AUGMENTED STRING PROTOTYPE

The augmented string instrument prototype consisted of a plastic, 14mm thick, optic fiber cable that was strung to a wooden structure (see Figure 1). Pure Data was used to process data and synthesize sounds based on incoming sensor data from an Arduino and a piezo element connected to a sound card. The sensors connected to the Arduino were an analogue 3-axis accelerometer (ADXL335) for measuring string displacement (placed on the top of the cable) and an ultrasonic rangefinder (LV-EZ4) for measuring the vertical position of the users' hand (placed next to the cable on the wooden structure, facing the floor). The 20 mm piezo element (7BB-20-6) was placed on the top of the cable to detect attack and velocity.

By striking or pulling and then releasing the string with a force above a certain threshold, the piezo element detected an "attack" on the string. Attacks were used to trigger a note. The volume, attack- and release time of the note depended on the force registered by the piezo element. The note sustain until the string had stopped vibrating, after about 300 ms. The accelerometer was used to sense changes in velocity along its axes due to slight displacement of the string, cause by either touching or shaking the string. Such actions caused the system to slowly fade in the previously played note with a volume depending on the level of the velocity. Keeping the velocity above a certain threshold (by for example continuously shaking or pulling the string) without triggering an attack on the piezo activated a wah-wah filter that increased in intensity the longer the velocity was above the threshold. If an attack was registered during these motions, the wah-wah filter was turned off.

At the moment of the attack, the distance from the top of the string to the hand (or other body parts closer to the sensor) was registered using the ultrasonic sensor. This distance was used to determine the pitch of the note on a major scale, spanning 3 octaves. The higher up the hand was placed on the string, the higher the pitch of the note. The force registered at the moment of the attack controlled three different types of sounds, each being an octave apart and with different characteristics. A small force triggered a low octave bass sound, a medium force triggered a middle octave clean sound and a stronger force triggered a higher octave chorus sound. If displaced sufficiently, the accelerometer could sense in which direction the string was moving, which for the low and the high sound was used to control a band pass filter. The frequency of the band pass filter was controlled by the direction of the angle of the vibrating string (0-360 degrees). A larger angle shifted the center frequency of the band pass filter towards a higher value. This effect was quite subtle due to the tightness of the string (especially at the point where the string was attached to a wooden frame), but provided a sweeping effect to the sound due to the string vibrating back and forth (between for example 0 and 180 degrees). This effect could also be achieved by dragging the string in a circle motion.

4.1 Limitations

There are several limitations of the tested prototype. The optic fiber cable prevented the use of sensors covering the string, as they would obstruct the emitted light. A crucial design challenge in this context was to sense the vertical hand position without using capacitive sensing or frequency detection of the strung string (the latter would be difficult since the plastic string vibrated with a very low frequency). Based on the above mentioned constraints, we opted for an ultrasonic sensor. Ultrasonic sound was chosen over infrared light for its longer range and to prevent instability due to changing light conditions at the museum exhibit. Unfortunately, the selected sensor turned out to be both inaccurate and unreliable in the interaction.

A future version of the string, to be deployed in early 2017, will include a LED light intertwined fiber optic cable with DMX controller, and a haptic floor which will be realized by placing a vibrating plate below each string. The vibrating plate will be activated through interaction with the string itself.

User	*Time*	*Pluck*	*Pull*	*Stroke*	*Shake*	*Strike (Finger)*	*Strike (Hand)*	*Mute*	*Flick*	*Box*	*Hold*	*Twist*	*Drag*
C1	2:00	•			•		Δ	•					
C2	4:50				Δ		Δ	•		•		•	•
C3	5:00				Δ		Δ			•		•	
Y4	3:00	•	Δ		Δ	•		•					•
Y5	7:30	Δ	Δ		•	•	•	•					•
Y6	6:30	Δ	•	•		Δ		•	•		•		
Y7	7:50	•	Δ	•	Δ	•	Δ	•			•		
P8	5:40	Δ	•	Δ	•	Δ	Δ	•	Δ				
P9	4:00	Δ	Δ	•	•	Δ		•			•		

Table 2. The participants' interactions with the prototype during the user tests. Participants' main modes of interaction are marked as triangles.

Figure 1. The augmented string instrument prototype. The optic fibre cable was not lit up during user tests.

5. RESULTS

We could identify a number of actions performed by our participants through analysis of the video material capturing the participant's interactions with the string, including: plucking, striking, shaking muting, flicking, boxing, holding, twisting, and dragging (see Table 2). We define these interaction types as follows. "Plucking" is defined as pulling and releasing the string with two or less fingers, while Pulling is defined as pulling and releasing the string with three or more fingers. "Muting" the string is holding the string to cancel its motion, while "holding" is defined as holding the string still and then releasing it. "Twisting" the string is turning the string around its own axis, and 'dragging" the string is pulling out the string without releasing it.

5.1 Modes of Interaction

All participants except the children initiated their interaction by plucking or pulling the string and maintained one of those interactions as their main mode of interaction. Two of the children, C1 and C3, initiated their interaction by plucking and poking respectively, but only once before moving on to other types of interactions, as opposed to C2 who immediately started shaking the string. The children were overall quicker when it came to starting hitting the string than the rest of the participants. They also struck with their hand rather than striking with their fingers (unlike the young adults and parents who did both). The children also used less variety in their way of interacting with the string, but instead they interacted in different ways than the other participants (e.g. boxing, twisting, heavy shaking and even hitting the string with their head). All children had shaking and striking with the hand as their main modes of interaction, while the young adults and parents mainly plucked, pulled or struck with their fingers on the string. All users except for C3 tried muting the string. Rare modes of interaction among the young adults and parents were flicking the string (done by Y6 and P8), stroking the string (done by Y6, Y7, P8 and P9), dragging the string (done by Y4, Y5 and also C2) and holding/releasing the string (done by Y6, Y7 and P9). It is also worth noting that none of the children pulled the string. See Table 2 for a more detailed overview of modes of interaction.

5.2 Concepts of the Instrument

The string metaphor was perceived differently among the participants. Users with previous experience of musical instruments (see Table 1) understood it as though the string would produce a sound when you touched it, while the rest of the participants instead believed it would start glowing (an expectation Y4 and Y5 also had). Almost half of the participants (C2, C3, Y5, P8) thought of a rope or a lace

instead of a string when they first saw it. After the initial interaction, some of the participants tried to produce different pitches by interacting at different heights of the string. Participants C3, Y5, Y6 and Y7 believed that the string would react like a piano or guitar with low pitch at the bottom and a high pitch at the top. The other participants instead discovered this property throughout the experiment, except for C1 who never found this property, and C2 who thought the pitch varied depending on the direction from which the string was hit. Y4 and Y5 felt that they could control the volume of the sound depending on the force they applied when striking the string, while Y7 expected it to be like that but did not feel the system responded in that way. Y5, Y6, Y7 and P8 were also curious whether the direction from which they hit the string had any effect on the sound.

The different types of sounds were noticed by all users, while this was not an expected behavior, many (C3, Y4, Y5, P8, P9) thought that it was exciting that there was more to discover. None of the users figured out how to control this aspect on their own and Y6 and Y7 expressed that this was confusing. Y7 also felt that triggering different sounds depending on the force caused the individual sounds to lack dynamics as it decreased the potential to adjust the volume. However, Y7 still found the potential of triggering different sounds interesting.

Y6 and Y7s perception of the instrument's complexity also differed from the rest. They felt that the instrument was very complex, with Y6 expressing that "it's usually easy to understand the concept of a new instrument", while the other participants perceived the instrument as "easy" because "you just need to touch it to make sounds".

The wah-wah filter was an appreciated element in the instrument as C1, P8 and P9 all uttered "Cool!" when they discovered it. Y7 also expressed that "this feels like I can control". Both C2 and Y4 also seemed to be in control of it, dragging it back and forth or shaking it several times, controlling the intensity of the filter. Y4, Y6 and P8 expressed the desire to be able to play more than one note or sound simultaneously on the string, in order to be able to play harmonies and not just a melody, or to be more than one person playing it. For that reason, C1, C2, C3, Y4, Y5 and P8 also wanted to have more strings, similarly to e.g. a harp.

5.3 Phases of Experience

overall, the participants expressed curiosity and excitement in the initial phase of the experience. Some (Y4, P8, P9) laughed for themselves while interacting with the string and others (Y1, Y3, Y4, P8, P9) uttered sentences like "this was cool" or "fun". Y5 and Y6 explicitly noted that the string was "very conspicuous, you just want to touch it".

After making the string produce sounds, the way of interacting with the string differed substantially between the children and the other participants. The children were noticeably intense in their interaction, using fast and energetic movements without much time for pauses or apparent reflections. The young adults and parents instead seemed more thoughtful and thorough in their approach, taking their time to reflect on their interactions and covered most of the more expected ways of interaction (as seen in Table 2). The children were mainly concerned with creating and discovering sounds. While C3 wanted to continue playing after the interview, C2 felt stated that "it was fun in the beginning, but then you got tired of doing the same thing all the time".

The adults (and C2) all tried but failed to control the string in order to successfully play a song or a melody. Y4 said that the pitch "felt random" and Y7 believed she was activating a predetermined loop. Y7 was particularly frustrated as her initial hopes to "become friends" with the string turned out to be difficult and she instead started to wonder whether "she was stupid or the string was stupid". Y6 also became irritated as "the string decided what note to play" and expressed weariness due to lack of control. In contrast, Y4, Y5, P8 and P9 all enjoyed playing the string and thought it was fun despite the unexpected lack of control. Although Y4 felt that the pitch was random he felt that "its cool that you can do so much with something as simple as touching a thing". Instead of blaming the instrument for lack of control, Y5, P8 and P9 expressed that if they had only been more musical they could probably have played it. P8 also felt that "it doesn't need to be so serious" in response to playing a melody, and that "its just cool to play around, even if you dont have control". P9 also said that the initial drive to just "play" soon evolved into a desire to play a song. For Y5, Y7 and P9 this became a problem that they wanted to solve, while Y7 got frustrated and desired a shorter "learning curve". In opposition, P9 felt that if it had been easier to play a melody she would have been "finished" with the installation quicker.

5.4 Interaction strategies

As seen in Table 2, the participants had several different ideas of interacting with the string. It was hard for the users to build further on these initial ideas of interaction due to the lack of control. Y7, for example, expressed that "Its hard to be creative when you dont have control over what notes you are playing" and that the string "lacked consistency". As mentioned previously, children showed less variation in their interaction than the adults, but instead interacted in different ways than the others.

As mentioned earlier, children were more intense and seemed less "careful in their interaction, hitting and shaking the string with more power compared to the other participants. Some young adults (Y4, Y7) felt that they dared to hit harder and interact in ways they probably wouldnt have with other string instruments. Y5 said it felt easier to hit and pull this string than other instruments, as "theres norms and rules for traditional instruments that dont exist for this one. However, some (Y5, Y7 and P8) did not dare to pull it out too much or shake it too hard in fear of destroying the instrument.

While most participants stated that they were too focused on finding out how the string worked to think about anything else, the installation triggered the imagination of some users, like those suggesting using more strings so that they could play the instrument like a harp. One user also said

that she felt like playing the string as an upright bass, or to have the string horizontally and play it like a piano.

6. DISCUSSION

The study was designed to investigate interaction with a large sized string instrument in a public setting. Although the prototype of the augmented string did not provide the reliability that was initially aimed for and affected the way users interacted with the string (in terms of what the users could control for as well as the user's expectations), the results still provide relevant insights into string interaction for museum settings.

The results support the ideas of layering the experience and allowing for different levels of engagement, as shown in previous studies [1–3, 5]. An augmented instrument in a public setting gives rise to particular design challenges; one needs to consider the age and musical background of the potential users, since these factors might affect both the interaction as well as the user's expectations. Depending on whom the experience will be designed for in first hand and what level of engagement that is desired for the particular exhibition, certain compromises regarding the instruments functionality might have to be done.

For initial engagement, an early success experience is crucial for maintaining interest in the exhibit [5]. This can be achieved by utilizing the affordances of the instrument. As the most natural affordances of a string is plucking and striking it, our string produced sound just by touching it. This property triggered immediate curiosity among the users. Such a property is an attribute that is referred to as "attractor" by Edmonds [22]. For prolonged engagement, the system needs to give the user the opportunity or desire to explore, experiment or challenge themselves, attributes that Edmonds refer to as "sustainers".

The augmented string offered elements of discoverability through different types of sounds and effects that could be triggered. The users could explore these functions by interacting with the string in various ways. The way our participants approached the string seems to depend on their musical background and age (or more precisely, the lack of certain experiences, rules and norms that you obtain as you get older). The children interacted with the string more intensely, while adults had a more thoughtful approach and at times stepped back from the instrument in order to reflect on their actions and the strings responses. The children's seemingly less reflective behavior can perhaps cause them to be guided by the design of the system, if they are continuously "rewarded" by a certain interaction.

With traditional string instruments, the volume is directly proportional to the amplitude of the strings vibrations, which can be dampened more easily when striking the string with the hand instead of plucking or pulling it. The risk of dampening the strings vibrations was not the case with our string as the volume instead was connected to the force applied by the participant when the piezo detected an attack. This, in combination with the lack of (or a different) conceptual model of how string instruments work, might be the reason to why none of the children pulled the string, and only one child plucked the string before quickly moving on to more intense interactions. This less reflective behavior is worth taking into consideration if a particular interaction is desired by the designer, and certain limitations might need to be set in order for children to not overlook "less-rewarding" interactions.

The childrens lack of certain behavioral rules and norms might also be the reason for interacting differently then the adults, such as boxing, kicking and hitting the string with their head. It might be important to consider the way children interacted with the string when designing for public installations. Seeing how the children were more focused on exploring than on completing a challenge (like playing a melody), can be important to provide discoverable functionalities and sound effects for their way of interacting in order to encourage prolonged engagement. Basic musical characteristics like duration, volume and timbre should perhaps be associated to more common modes of interaction as a way to keep the user's explorative journey moving forward to the next levels of engagement; experimentations and challenges.

It was more obvious how the adults were more systematic in their explorations than the children, especially among those with more musical experiences, who for example expected a different pitch at different heights of the string (associating it with a guitar or a piano), or that the direction they hit the string from should affect the sound. The most common type of experimentation among the adults was trying to achieve the same note by hitting the string at the same place or with the same force, but instead it yielded unexpected results. The children were also seen hitting the string at the same place consecutive times, but perhaps for a more exploratory reason due to its "randomness" (being "rewarded" with a new sound with almost every strike), as none of them explicitly tried to control the sound in that way in order to play a melody (unlike all the adults). Some users noted a difference in volume depending on the force applied, but the correlation was unclear. This was probably because the force also triggered different sounds, thereby also disturbing the sounds perceived dynamics.

The most commonly expressed challenge among the participants was, as previously mentioned, to play a melody or a song. This was probably due to the augmented strings natural associations with traditional string instruments, leading to natural expectations of being able to play a melody. Some users also thought that many notes and sounds could be played simultaneously, probably due to the same associations. Although the string failed to meet these expectations, most of the participants still felt they had a fun experience of exploring and experimenting. Some even blamed themselves for these shortcomings of the string by assuming that they did not have the musical skills needed to play melodies. On the other hand, those who had musical background and experience of playing instruments, became frustrated and irritated by the strings shortcomings.

Although we used sensors with good resolution, the sensing methods were not reliable enough to provide a responsive and expressive experience. While it is difficult to talk about originality in regards to the users ideas of interaction, due the low number of test participants and the study setup,

it is notable that none of the participants felt that they could create original music with a clear melodic structure. It was especially the uncontrollable pitch that users felt prevented creativity, as expressed by one of the more musically experienced users, it "was the main obstacle for being creative".

A refinement of the prototype is required in order to make it more controllable. For instance, providing a way to control pitch is vital to support creating melodies. This could be achieved by using other sensing methods or evaluating other types of interactions suitable for controlling pitch. Pitch control could also be omitted from the instrument, which could instead provide an explorative soundscape. However, this might not create an equally engaging experience for musically experienced users. Implementing a predetermined melody loop might also be an option, but causes the instrument to lose some of its open-ended qualities.

One of the most appreciated and engaging elements of the instrument among the participants was its discoverability. This element should definitely be retained by providing exploratory sounds or effects for less common interactions (like kicking and heavy shaking), while keeping fundamental functionalities (like volume and/or pitch) to common ways of interactions, unlike the prototype presented in this paper where changing the type of sound interfered with the control of the notes volume.

Acknowledging that interactive exhibits are especially attractive to children and their families [6], along with our result that demonstrates substantial differences in how adults and children approach a string, particularly stresses the need to consider age when designing interactive exhibits in public settings.

7. CONCLUSION

In this study we have looked at how people interact and experience a large size augmented string instrument, intended for a museum installation.

The explorative elements of the installation proved to be the most engaging among children, therefore it seems important to design discoverable functionalities for their intense and sometimes unconventional ways of interacting with the string. The self-imposed and more traditional challenges created by adults (e.g. playing a melody) need to be treated in a satisfactory but balanced way, in order to meet them.

Previous research has highlighted the importance of layering different levels of engagement, to prolong use, in installations in public settings. Our study adds to this by highlighting that we also need to account for different age and musical backgrounds when designing these kinds of installations. Furthermore, to make sure that users dare to explore and experiment with the installation, it is important that it is perceived as stable, controllable, and enduring.

Acknowledgments

This project is partially funded by a grant to Roberto Bresin by KTH Royal Institute of Technology, and by Musikverket - Scenkonstmuseet/Swedish Museum of Performing Arts.

8. REFERENCES

[1] K. Liu, "Designing visitor experience for open-ended creative engagement in art museums : A conceptual multi-touch prototype design," p. 165, 2013.

[2] S. Allen and J. P. Gutwill, "Designing science museum exhibits with mulitple interactive features: Five common pitfalls," *Curator: The Museum Journal*, vol. 47, no. 2, pp. 199–212, 2004.

[3] Z. Bilda, E. Edmonds, and L. Candy, "Designing for creative engagement," *Design Studies*, vol. 29, no. 6, pp. 525–540, 2008.

[4] P. Campos, M. Campos, J. Pestana, and J. Jorge, *Studying the role of interactivity in museums: Designing and comparing multimedia installations*, 2011, vol. 6763 LNCS, no. PART 3.

[5] E. Hornecker and M. Stifter, "Learning from interactive museum installations about interaction design for public settings," *Proceedings of the 20th conference of the computerhuman interaction special interest group CHISIG of Australia on Computerhuman interaction design activities artefacts and environments OZCHI 06*, p. 135, 2006.

[6] J. Kidd, I. Ntalla, and W. Lyons, "Multi-touch interfaces in museum spaces : reporting preliminary findings on the nature of interaction," *Computer*, pp. 5 – 12, 2011.

[7] G. Taxen, S. O. Hellström, H. Tobiasson, M. Back, and J. Boewrs, "The Well of Inventions Learning, Interaction and Participatory Design in Museum Installations," in *Seventh International Cultural Heritage Informatics Meeting*, 2003, pp. 8–12.

[8] B. Knichel and P. Kiefer, "Resonate a Social Musical Installation Which Integrates Tangible Multiuser Interaction," pp. 111–115, 2015.

[9] F. Bevilacqua, N. Rasamimanana, E. Fléty, S. Lemouton, and F. Baschet, "The Augmented Violin Project: Research, Composition and Performance Report," in *Proceedings of the 2006 International Conference on New Interfaces for Musical Expression (NIME-06)*, vol. 9, 2006, pp. 402–406.

[10] T. Grosshauser and T. Hermann, "New Sensors and Pattern Recognition Techniques for String Instruments," in *New Interfaces for Musical Expression (NIME)*, no. Nime, 2010, pp. 271–276.

[11] T. Grosshauser and G. Tröster, "Finger Position and Pressure Sensing Techniques for String and Keyboard Instruments," in *Proceedings of the International Conference on New Interfaces for Musical Expression*, 2013, pp. 479–484.

[12] E. Guaus, T. Ozaslan, E. Palacios, and J. L. Arcos, "A Left Hand Gesture Caption System for Guitar Based on Capacitive Sensors," in *NIME 2010 Proceedings of the*

International Conference on New Interfaces for Musical Expression, 2010, pp. 238–243.

[13] A. P. McPherson, A. Gierakowski, and A. M. Stark, "The Space Between the Notes: Adding Expressive Pitch Control to the Piano Keyboard," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, 2013, pp. 2195–2204.

[14] A. McPherson and Y. Kim, "Augmenting the Acoustic Piano with Electromagnetic String Actuation and Continuous Key Position Sensing," *Signal Processing*, pp. 217–222, 2010.

[15] D. Newton and M. T. Marshall, "Examining How Musicians Create Augmented Musical Instruments," in *Proceedings of the 2011 International Conference on New Interfaces for Musical Expression (NIME2011)*, no. June, 2011, pp. 155–160.

[16] H. Tobise, Y. Takegawa, T. Terada, and M. Tsukamoto, "Construction of a System for Recognizing Touch of Strings for Guitar," in *Proceedings of the International Conference on New Interfaces for Musical Expression*, 2013, pp. 261–266.

[17] J. Dewey, "Art as Experience," in *Art and its significance: An Anthology of Aesthetic Theory*, 1994, pp. 205–220.

[18] J. McCarthy and P. Wright, "Technology as Experience," *Interactions*, vol. 11, no. 5, pp. 42–43, 2004.

[19] G. S. Morson and C. Emerson, *Mikhail Bakhtin: Creation of a Prosaics*, 1990, vol. 44, no. 4.

[20] R. J. Sternberg, K. Sternberg, and J. S. Mio, *Cognitive psychology*. Australia; Belmont, CA: Wadsworth/Cengage Learning, 2012.

[21] F. E. William, "The cognitive-affective interaction model for enriching gifted programs," in *Systems and models for developing programs for the gifted and talented.*, 1993, pp. 461–484.

[22] E. Edmonds, "On creative engagement," *Visual Communication*, vol. 5, no. 3, pp. 307–322, 2006.

A LIBERATED SONIC SUBLIME: Perspectives On The Aesthetics & Phenomenology Of Sound Synthesis

Anders Bach Pedersen
IT University, Copenhagen, Denmark
anders@tbc.dk

ABSTRACT

In this paper I will investigate the aesthetics of electronic sound synthesis, materiality and the contemporary sublime in an analysis and discussion of interrelated phenomenological, philosophical and cultural considerations through chosen sound and music examples. I argue that the aesthetic experience of sonic timbres that seem unearthly to us resembles that of a transcendental sublime in the uncanny experience of the synthesis of both known and unknown sounds. Both experimental music and "switched-on" reinterpretations are addressed through explorations of sound in time, space and technology and I discuss if we as listeners are able to differentiate materiality from its superficial cognates when challenged by sonic doppelgängers. Concepts of sonorous perception are taken into account from a phenomenological point-of-reference with the purpose of discussing a Varèsian liberation of sound synthesis, arguing the transcendence of the boundaries in the physical world being possible through the aesthetics surrounding an unfathomable technological sublime in the art and infinite sea of possibilities of synthesizing electricity.

1. INTRODUCTION

In the creation of an electronic sound, analog or digital, the synthesized sound can, roughly speaking, be described as either resembling a natural sound, e.g. a physical model or something from nature itself, or the sound can be non-specific and its liking non-existent in the physical world. It serves this paper to use the descriptions made by Steven R. Holtzman [1] about these types of sound synthesis being either "standard" or "non-standard", alongside John Chowning's [2] notion on "known" or "unknown" timbre and my own definitions of "familiar" and "unfamiliar" sounds. This division in synthesis methodology in many ways resembles when we as listeners react and try to put into words the aesthetic experiences of listening to electronic sonic timbres: We can either relate the sounds to something we know from the natural world or our traditional ecological knowledge [3] of music, or we can ultimately try to fathom the sounds in their own existence and in conjunction with one another to provoke a mood, emotional and/or sensory relation within the listener. The objectification and paradox in the effort of materializing sound becomes a personal aesthetic judgment of timbre and Kantian aesthetics [4], but I will later in this paper argue that it is in fact an active participation in a phenomenological *liberation of sound* in continuum of the idea presented by Edgard Varèse [5] in 1936. I do acknowledge specific types of synthesis – additive, subtractive, FM etc. – with regards to their different timbral qualities, albeit theory in this area does not serve any major importance for the points made in this discussion as it is mainly based on aesthetics.

In this paper I will firstly discuss the nature of sound synthesis in connection with Freud's [6] notion of *the uncanny* through a phenomenological and sensory-based viewpoint in the relation between perception and the ambiguous materiality of sound synthesis. I will elaborate on these concepts in connection with the aesthetic experience as presented by Goldman [7] through various theoreticians and philosophers, and later with the contemporary sublime [8]. All discussions will be exemplified through examples of chosen sound and music that benefit the argument of this article in the case of aesthetics and not in depth in relation to music theory, psychoacoustics or mathematics, although I acknowledge the importance of these factors. Finally I will reflect on the topic by discussing the significance of a phenomenological revisit to Varèse's *liberation of sound* in the experience of both 20th and 21st century electronic sound synthesis and music through modern aesthetics and the contemporary sublime.

2. SONOROUS PERCEPTION, AESTHETICS & SPATIALITY

There is a fundamental phenomenological interest present in the discussion of the synthesis of familiar and unfamiliar sounds in terms of sensory perception that needs to be addressed from its origin. In his book, "Phenomenology of Perception", Maurice Merleau-Ponty [9] makes a distinction between what he (through the terminology of neurologist Kurt Goldstein) calls *zeigen* and *greifen* [ibid.,] (p. 116), meaning roughly the movements of *pointing* and *grasping*. He hereby assesses the differences in perception in terms of vision and hearing, and argues that:

"[…] sound, of itself, calls forth […] a grasping movement, while visual perception calls forth a designating gesture." [ibid.] (p. 116)

The idea is that sound is something we connect to the movement of grasp, i.e. tactile cognition – what one through Merleau-Ponty's notions of "sonorous stimuli" and "auditive sensitivity" [ibid.] might define as a type of *sonorous perception*, much alike Juhani Pallasmaa's [9] dividing of vision and hearing into experiences of *exteriority* and *interiority*. When we make an aesthetic judgment of music or sound we tend to use these references of tactility, something interior: A sound can be perceived as hard or soft, coarse or delicate, cold or warm etc. In the realm of musical metaphors perceptual signifiers, as Back [11] argues, are some of the only culturally based descriptive means in terms of moving towards an abstract language in music, as "The step from abstraction to abstract representation has not yet fully occurred […]" [ibid.] (p. 164). Of course, in the experience one also relates music, as with any art form, to a certain feeling or mood, but, as Alexander Baumgarten [7] would argue, the aesthetic is first and foremost "[…] cognition by means of senses, *sensuous knowledge*." (p. 255, my italics). This conclusion corresponds through its almost empiristically epistemological sense with Merleau-Ponty's concepts of sonorous perception in terms of the experience of sound in oneself, in the spatiality of one's own body and one's psychoacoustics:

"Sound always directs us towards its content, its signification for us; in visual presentation, to the contrary, we can much more easily 'abstract' from the content and we are much more oriented toward the location in space where the object is situated." (from Goldstein's (1931) "Über Zeigen und Greifen", cited in [9] p. 116)

In the realm of space and time in terms of sound appearing in the world, one could argue that sound we *know* from and sensory perceive in the natural world, as opposed to a synthetic sound, appears in an acoustic environment in time and therefore exists. The spatiality of one's own body and the spatiality of the physical world are perceptually inseparable, but the distinction between the two corresponds in certain ways with when we, in the experience of sound, relate what we *hear* to what we *know*; We expect the noise of the world to resonate with the noise of the body. This is evident in the medium of recording technology, which is arguably a method for remediation and the manipulation of spatio-temporality for a repeated and identical experience via playback. Here follows an example:

In 1937 Olivier Messiaen premiered his piece "Oraison" [12], also known as the fifth movement of the "Fête des Belles Eaux"; a site specific composition played from tape through loudspeakers at an exhibition of fireworks along the Seine in Paris [13]. The piece was written for one of the earliest electricity-based instruments, the Ondes Martenot [14]. The Ondes Martenot emits a ghostly, unearthly sound and shares the timbral characteristics of a string instrument like, for instance, the cello. In 1941 Messiaen re-wrote the piece into a movement for his chamber work "Quatour Pour la Fin du Temps" called "Louange à l'Éternité de Jésus" [15]. In this version, the electronic sounds of the Ondes Martenots used for "Oraison" were replaced by acoustic piano and, in fact, a cello.

The recordings listed in the references of this paper exemplify the difference in spatiality as an active in the aesthetic and sensory experience of sound. These experiences of recorded sound are anchored in phenomenological and cultural heritage. We will begin by asserting the former: In the recording of "Louange à l'Éternité de Jésus" it is apparent that the listener is positioned in church- or concert hall-like acoustics that reverberate the sounds emitted by the instruments in a delicate manner. The piece is, more or less, written for these particular performance spaces and not initially for recording. In the case of "Oraison", although it is the exact same piece compositionally, the listener has no point-of-reference in the beginning of the piece as to which space we as listeners are situated in or even what instrument is in fact being played, and therefore whether or not the piece is even performed by a human being. The same perceptual considerations might have been evident at the premiere of the piece as it was played back from tape at the Seine in 1937 [13] – here both the medium, tape, and the instrument, the Ondes Martenot, become transmitters of an unearthly sound.

The perception of a recording of the Ondes Martenot as a musical instrument, and its sonic relatability to e.g. the cello, also stems from its embedding in the listeners traditional ecological knowledge (hereafter TEK) [3]. Berkes et al. define TEK as the "[…] cumulative body of knowledge, practice, and belief […]", "[…] an attribute of societies with historical continuity in resource use practice." (p. 1252) Despite the main focus on local ecological/environmental knowledge and corresponding resource use activities, the anthropological notion of TEK serves the cultural and historical argument of this paper. The distinction between known and unknown sounds is rooted in the TEK of Western music tradition. The Ondes Martenot has no instant recognizable timbre because of its limited time and use in the continuity of Western music tradition and cultural heritage. The "hollow" or "nasillard" [13] timbre, whether in the context of classical music like "Oraison" or contemporary pop like Daft Punk's "Touch", remains sonically abstract to this date.

In this exemplification the sounds that are known to us by music tradition – in "Louange à l'Éternité de Jésus", the cello and the piano – present a cultural and perceptual frame of reference that is immediate and materially related to the sense of sight, "the location in space to where the object is situated." [9] (p. 116) Although "Oraison" compositionally predates "Louange à l'Éternité de Jésus", the aesthetic experience provokes a response of wonder due to the sound of electronic synthesis; even though it timbrally relates to the cello, the Ondes Martenot is still a sound we will have difficulties *grasping*, because sounds generated by oscillators exist in a non-spatial, all temporal environment; They are essentially acoustically bound by nothing. The concepts of sonorous perception of Merleau-Ponty and Goldstein in terms of sound directing us

toward *content* rather than *space* or *location* [ibid.] hereby becomes an essential angle in the investigation of the phenomenology and aesthetics of sound synthesis, whether the sound is part of a TEK of music or not. I will later discuss if in fact the only way to indeed grasp electronic sound synthesis is through an aesthetic experience related to a transcendental sublime.

3. THE UNCANNY MATERIALITY OF ELECTRONIC SOUND

To further investigate through the aesthetic angle on sound synthesis methodology, I will relate the concept of *the uncanny*. In 1919 Sigmund Freud wrote an essay in which he described the phenomenon of "Das Unheimliche" [6] – as in the opposite of "heimlich", literally meaning the "un-homely" – popularly translated as "the uncanny". It is from Freud's point of view related to aesthetics being the "[…] qualities of feeling." [ibid.] (p. 217) The *feeling* of the uncanny appears when something familiar simultaneously seems unpleasantly or strangely unfamiliar resulting in an emotional response of fright and/or wonder. The Japanese robotics professor Masahiro Mori returned to the subject in 1970 from the viewpoint of technology with his article about "The Uncanny Valley" [16]. He related the uncanny to the human likeness of robots, puppets and zombies, but also to the same considerations as Freud in terms of animate and inanimate beings.

As mentioned in the case of Messiaen's "Oraison" [12], the ghostly, hollow sound of the Ondes Martenot instrument leaves the listener with an eerie sensation although timbrally it is much alike an acoustic string instrument. The sounds are familiar yet strange, a kind of unsettling re-interpretation rather than a direct emulation of a well-known acoustic instrument. In the case of the vocoder [14], we are explicitly dealing with sounds generated solely from a human being synthesized into something that can be experienced as being uncanny. In the case of Bruce Haack's "Electric To Me Turn" [17] the vocoder represents the most explicitly uncanny, robotic quality of a purely electronic take on a musically Western and fairly traditionally composed blues tune. The vocoder comes to represent something familiarly unfamiliar and perhaps even ominous, which is evident in the title of Haack's album: "Electric Lucifer" [ibid.] – an electronic demon within.

The re-interpretation/-synthesis of something essentially human-made or human-like formed the use of sound synthesis at the beginning of its utilization in the arts and in music in the 20th century. Historically the voyage into uncanny electronic synthesis of familiar sounds is far-reaching within the commercial world of both instrument-design of synthesizers (e.g. the Yamaha CS-80 or later the DX7) and in music; most notably through the Moog synthesizer "switched-on" reinterpretations of famous classical works as executed by Wendy Carlos (with the album "Switched-On Bach" [18]) and Isao Tomita (with the album "Snowflakes Are Dancing" [19]). The syntheses of sounds we *know* from the physical world were naturally more useful from a commercial viewpoint, than the strange and incomprehensible, intangible noise-like *unknown* sounds of the experimental avant-garde from the same era due to missing perceptual [11], but also material signifiers in the general TEK [3] of Western music.

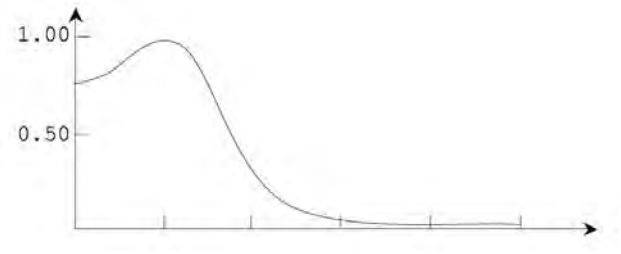


Figure 1. Drum envelope in Chowning [2], p. 8.

Perhaps unalike in commercial usability, standard and non-standard syntheses [1] share the attributes of a certain ambiguity in materiality and I argue that both these methods are essentially uncanny in the referential space of the natural world. Bill Brown [20] deals with the concept of materiality and argues that even though "[…] the material serves as a commonsensical antitheses to […] the spiritual, the abstract, the phenomenal […]" "[…] *materiality* has a specificity that differentiates it from its superficial cognates, such as physicality, reality, or concreteness." [ibid.] (p. 49) He references the assertion of materiality as to something's "[…] look and feel, not simply its existence as a physical object." [ibid.] Indeed, when talking about materiality, one must first and foremost either acknowledge an object on a sensory basis, much like the aforementioned *zeigen* and *greifen* [9], or try to assert something's immateriality, which in the extreme cannot be experienced phenomenologically.

However, in the case of sound generated by electricity, asserting Brown's materiality becomes somewhat a vast space of possibilities of *materialization* more than a distinct materiality or immateriality. The materiality of sound synthesis is therefore somewhat equivocal and uncanny in the course of tangibility. A familiar sound, e.g. of the form (envelope, fig. 1), resembling a drum, as can be heard in Laurie Spiegel's "Drums" [21], is via a listener's objectified and material-focused perception a natural drum, but from an aesthetic point-of-view it becomes something else, something known yet unknown and indeed uncanny. It is and is not the referenced object, an acoustic drum, in part because of its derived relation to any natural spatiality as mentioned earlier – it becomes in a sense a sonic *doppelgänger* [6]. As for synthesized sounds that are initially unfamiliar, in e.g. Morton Subotnick's classic "Silver Apples of the Moon" [22], we are offered no point-of-reference in the natural world whatsoever to the origin of what we are listening to. The same can be said in the case of Messiaen's "Oraison" [12], but in "Silver Apples of the Moon" the sounds we are presented with leave us in a state of even greater wonder in terms of music cognition or naturalistic relation. Our closest metaphorical parallel to materiality then arguably becomes the albeit erroneous theory of the *fluid* matter of electricity as introduced by Benjamin Franklin [23];

throughout electronic sound synthesis methodology, sound flows freely from one form to the next. We are then left only with an aesthetic judgment of taste, in a Kantian sense free of acoustic "interest" [4]: Are we in fact aesthetically pleased with what we hear, or will we persistently try to sonorously reference these unearthly sounds to any material known to us from the physical world and our cultural heritage? When even the doppelgängers become alien and even more so unrelated to our material reality, do we in fact then, in reference to Brown [20], differentiate materiality from its superficial cognates?

What remains is that the listener can neither *point* out nor *grasp* these electronic sounds [9]; whether familiar or unfamiliar, we cannot *see* nor *feel* them [20] – no matter if they are created through standard or non-standard sound synthesis. At the same time, the sounds are, through recording media, within our sonorous perceptive reach because of our constant reference to the world in which we encounter this aesthetic experience. The experience of familiar electronic sounds is thereby essentially the same as that of the unfamiliar, making both types of synthesis alike in their respective uncanny, unnatural materiality. The ever-morphing matter of electricity becomes in a sense phenomenologically perceivable by the electronic synthesis of sound. A self-defined "uncanny valley" [16] of these unearthly sounds could arguably become a representation of the interrelation between affinity and, instead of human likeness, likeness of the natural world, of acoustics, even towards the laws of physics.

4. AN AESTHETIC LIBERATION OF REMATERIALIZED SOUND

A paramount argument in this paper lies in the realm of an aesthetic experience both sublime and phenomenological in nature. It is necessary to acknowledge the theory of the contemporary sublime and, first and foremost, Edgard Varèse's lectures on "The Liberation of Sound" [5] collected in 1966 from transcripts spanning from 1936 to 1962. Varèse talks of the boundaries of traditional musical instruments and of linear musical counterpoint and how he foresees "New Instruments and New Music" [ibid.] (p. 17) to allow the writing of a clearly perceivable "[…] movement of sound-masses, of shifting planes." [ibid.] "The liberation of sound" exists, among other things, in what he calls "Music as an Art-Science" [ibid.] (p. 19) in which the medium of expression is a "sound-producing machine", in this paper regarded as opposed to the re-production known at the time from phonographs. In a lecture he concludes this new medium to be electronic [ibid.] (p. 19-21).

The contemporary sublime [8] shares from a perceptual perspective many attributes with the traditional sublime: It is an experience that provokes a lost-for-words response, a mute encounter with "[…] intimations of otherness or infinity." [ibid.] (p. 12), in the Kantian sense of the word provoking reactions of awe because of overwhelming size and magnitude, force or because of something beyond material existence or ordinary perception (a transcendental sublime (although not directly mentioned as such by Kant)) [4]. The sublime experience in many ways lies on the threshold of wonder and horror, mixing sensations of delight and fear [8]. Whereas 18th century sublime largely dealt with nature to instill the awe-inspiring experience, the source of wonder in the contemporary sublime is "[…] the incredible power of technology." [ibid.] (p. 12) Varèse, although not frightened per se, was like many thinkers of early and mid 20th century [ibid.] (p. 17) arguably in a state of continuous uncanny or sublimity due to the booming technological progress of his time.

Here follows a discussion of and an aesthetic approach to Varèse's "liberation of sound" through the unity of the topics that hitherto have been dealt with: perception, spatiality, materiality and the uncanny resulting in a sublime aesthetic experience. Firstly it is important to note that uncanny sounds have always provoked some kind of emphatic response, most notably from a historic viewpoint in the 20th century via the Italian futurist movement: The audience attending Rusollo's infamous noise-concert premiere in Milan in 1914 allegedly started a bloody riot [24]. This violent reaction, whether being provoked by societal conditions or physical discontent, is a powerful example of how noise, the aesthetic judgment of "[…] any sound one doesn't *like*." [5] (p. 20, my italics), can result in a thundering physical response. People experience horror in the unknown and are bewildered, lost-for-words and even frightened – the audience of Rusollo's concert indeed had a sublime experience of the uncanny in noise. As mentioned before, traditional ecological knowledge [3] guides our sonic perceptual signifiers [11].

However, I do not believe the experience of electronic sound synthesis to be initially sublime. I believe that it is possible through the lens of phenomenology to open certain doors towards a sublime experience by revisiting Varèse's "liberation of sound" with the very sounds he discussed; to open certain doors towards freeing electronic sound from a superficial perceptual and cultural categorization. A sonic sublime of sound synthesis begins with the acceptance of this aforementioned non-spatial, interior [10] frame of reference; that the forms we contain in standard (even in non-standard) synthesis [1] are derived from a continuous current: a Pandora's Box that we open and close by means of changes in e.g. amplitude, the envelope of a sound. It is when we let it out in the world, when we let the electricity flow through the ether surrounding the body [23], when we amplify the sound, that it becomes a part of the natural world of acoustics. However, when contained, the current that is our electrical signal still runs, still exists when powered, as a current in time but not in space.

While the sublime deals with uncontainable forms, materiality [20] has yet again to be taken into account in terms of the aesthetics of sounds in the contemporary experimental electronic avant-garde. Autechre's audio-visual "Gantz Graf" [25] from 2002 still remains one of the most interesting examples of the uniting of the senses, a unity in perception (*zeigen* and *greifen* [9] in particular) in an experiment of objectification and rematerialization. In the video, Autechre plays with the idea of the object

(fig. 2) as the center of visual and auditory *dematerialization* [20]. In this example we are in a very explicit way dealing with all of the aforementioned sonorous perceptive [9] specificities: Uncanny [6] sounds, both familiar and unfamiliar, that in complementing linearity resemble rhythmic, harmonic and melodic instrumentation and musical functions, have been *dematerialized* only to be audiovisually *rematerialized* [20]. Through the visual representation of an uncanny object, Autechre suggests an aesthetic that is beyond references to natural acoustics and visual space: The piece opens up for immediate transcendence into the arts from something sonorously beyond the perceptive compass of the physical world. The matter of electricity becomes an instrument for sonic materialization.

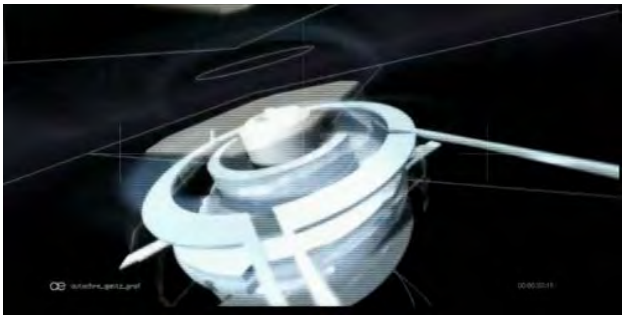


Figure 2. Autechre's "Gantz Graf" video (2002)

Where Autechre [25] represents dematerialization through "Gantz Graf" in a very high-paced, almost hyperactive manner, the same considerations towards the phenomenological and aesthetic aspects of this sonic sublime also exist in more slowly evolving, drone-like music. John Chowning's "Stria" [26] deals with the frequency modulation (FM) synthesis that he himself invented. With FM you can alter, as described in Chowning's research paper [2], the character of temporal evolution to resemble *known* natural timbres. But this technique also paved the way for Chowning to construct otherworldly and uncanny sounds without initial natural spatiality or relation to TEK [3]. The fact that techniques and ideas for natural resemblance can spawn unnatural sonics is evident in Eliane Radigue's "Triptych" [27] as well: a work dedicated to the five elements, where e.g. synthetic noise is used and filtered to create the sound of wind, but in the process becomes something else. The sounds of a conceptual containment and interpretation of an awe-inspiring nature becomes an entirely different aesthetic experience than a mere emulation or recording of the same phenomena.

The fact remains: Varèse's "sound-*producing* machine" [5] (p. 19), the technology that is the source of the sonically sublime experience, is a reality and is as important for the aesthetics of the matter as the resulting art. Through this creation of otherworldly timbres music truly becomes an "Art-Science" [ibid.]. This is where my discussion differs from Pierre Schaeffer's idea of "Acousmatics" [28]. "Acousmatics" refers to a phenomenological notion of separating sound from source in reference to an old Pythagorean teaching method in which the teacher lectures behind a veil, hidden from the line of sight of the students leaving them only with the sound of their master's voice. Despite the fact that Schaeffer's benchmark is in the manipulation of tape (i.e. *musique concréte*), there are phenomenological likenesses to this paper in terms of the sensory-based experience of listening as well as disregarding the Cartesian dualistic mind/body-argument. The difference is in the discussion of aesthetics. In this paper the aesthetic experience appears in the rejoining of sound and source in the materialization that electricity affords much more than the phonographic medium that Schaeffer refers to. The spatiality of recordings is, although manipulative, initially identical on each playback, whereas the spatiality of electricity is non-existent – sound synthesis is initially all about temporality. However, "instrumental progress" [28] (p. 81) is central no matter viewpoint.

Neither a separation of *zeigen* and *greifen* nor of object or subject in the perceptual identification of material are imperative towards a contemporary sonic sublime experience. If we instead sonorously perceive and accept the rematerialization of something as literally invisible and ungraspable as electricity, we are rewarded with a profound aesthetic experience. The notion of the machine is as important as the art itself if we can accept the new material, the new sounds, the "New Instruments and New Music" we are presented to. We can in a sense through the aesthetic experience of sound *emancipate* [4] the uncanny material – if we dare.

5. CONCLUSION

In this paper I have tried to map an aesthetic trail between known and unknown sounds in electronic sound synthesis. I have done so by means of phenomenological and aesthetic philosophy and theory concluding that the aesthetic experience of otherworldly electronic sounds is, however related to a traditional ecological knowledge of music, dependent on the unity of sonorous perception, spatiality, the materiality, de- and rematerialization of sonic substance. The concept of the uncanny is used to underline and illustrate the tension between familiar and unfamiliar sounds in connection with the aesthetic perceptive experience. It emphasizes the ambiguous materialization offered by electricity in both standard and non-standard synthesis, that from a phenomenology of perception are alike.

Sound and source are inseparable, like the interior and exterior spatiality of the body and the world. But the "sound-*producing* machine" is aesthetically sublime: It is the source to something beyond the Pythagorean veil, an uncanny release and movement of sound-masses from Pandora's box. However, nothing in the realm of sound synthesis is veiled; the matter of electricity is initially invisible and intangible. Because of this the transcendental sublime is a means to truly aesthetically experience and liberate the unearthly timbres of electronic sound from their immediate exterior parallels.

Acknowledgments

I wish to thank Susana Tosca and Hanne-Louise Johannesen from the IT University in Copenhagen for guidance in the writing of this paper and the surrounding research, and the reviewers from SMC 2016 for detailed and sound advice.

6. REFERENCES

[1] S. R. Holtzman, "A Description of an Automatic Digital Sound Synthesis Instrument" in DAI research report No. 59. Department of Artificial Intelligence, 1978, pp. 53-61.

[2] J. M. Chowning, "The Synthesis of Complex Audio Spectra by Means of Frequency Modulation" in J. Audio Eng. Soc. 21, 7, 1973.

[3] F. Berkes, J. Colding & C. Folke, "Rediscovery Of Traditional Ecological Knowledge As Adaptive Management", in Ecological Applications 10(5). The Ecological Society of America, 2000, p. 1251-1262.

[4] I. Kant, "Critique of Aesthetic Judgment" in Critique of Judgment, 1790, trans. J. C. Meredith. The University of Adelaide, 1911.

[5] E. Varèse & C. Wen-Chung, "The Liberation of Sound", 1966, in C. Cox and D. Warner (Eds.), Audio Culture: Readings In Modern Music. Continuum International Publishing Group, 2004, pp. 17-21.

[6] S. Freud, "The Uncanny", 1919, in J. Strachey (Ed.), The Standard Edition of the Complete Psychological Works of Sigmund Freud, Volume XVII (1917-1919): An Infantile Neurosis and Other Works. WW Norton & Company, 1976, pp. 217-256.

[7] A. Goldman, "The Aesthetic", 2001, in B. Gaut & D. M. Lopes, (Eds.), The Routledge Companion to Aesthetics. Routledge, 2013, pp. 255-266.

[8] S. Morley, "The Sublime", in Documentary of Contemporary Art. Whitechapel Art Gallery, 2010.

[9] M. Merleau-Ponty, Phenomenology of Perception, 1945, trans. D. A. Landes. Routledge, 2014.

[10] J. Pallasmaa, The Eyes of the Skin: Architecture and the senses. Wiley-Academy Press, 2005, pp. 39-80.

[11] M. Back, "The Reading Senses: Designing Texts for Multisensory Systems", in G. Liestøl, A. Morrison and T. Rasmussen, Digital Media Revisited. The MIT Press, 2003.

[12] O. Messiaen, "Oraison", 1937, in An Anthology Of Noise And Electronic Music, Vol. 4, LP. Sub Rosa, 2006

[13] Atma Classique, "Messiaen: Fête Des Belles Eaux." [Online]. Available: http://www.atmaclassique.com/En/Albums/AlbumInfo.aspx?AlbumID=360 [Accessed: May 14, 2015]

[14] M. Vail, The Synthesizer. Oxford University Press, 2014.

[15] O. Messiaen (1941). "Louange à l'Éternité de Jésus", 1941, in Quatour Pour la Fin du Temps, LP. Harmonia Mundi, 2008.

[16] M. Mori, "The Uncanny Valley", 1970, trans. MacDorman, K. F. and Kageki, N. in IEEE Robotics & Automation Magazine, June 2012, 2012.

[17] B. Haack (1968). "Electric To Me Turn", 1968, in Electric Lucifer, LP. Columbia Records, 2007.

[18] W. Carlos, "Air on a G String" in Switched-On Bach, LP. Columbia Records, 1968.

[19] I. Tomita, "Suite Bergamasque: Clair de Lune, No. 3." in Snowflakes Are Dancing, LP. RCA Red Seal Records, 1974.

[20] B. Brown, "Materiality" in W. J. T. Mitchell and M. B. N. Hansen (2012) (Eds.), *Critical Terms for Media Studies*. The University of Chicago Press, 2012, pp. 49-63.

[21] L. Spiegel, "Drums", 1979, in The Expanding Universe, LP. Unseen Worlds / Philo, 2012.

[22] M. Subotnick, "Silver Apples of the Moon, Part A", 1968, in Silver Apples of the Moon, LP. Digital Music Digital, 1994.

[23] K. Veronese, "Benjamin Franklin's Fluid Theory of Electricity" [Online]. Available: http://io9.gizmodo.com/5923754/benjamin-franklins-fluid-theory-of-electricity [Accessed: July 5, 2016]

[24] B. Thorn, "Luigi Rusollo (1885-1947)" in Sitsky, L. Sitsky, Music of the Twentieth-century Avant-garde: A Biocritical Sourcebook. Greenwood Press, 2000, pp. 415-420.

[25] Autechre, i.e. S. Booth & R. Brown, "Gantz Graf" in Gantz Graf, EP. Warp Records, 2002.

[26] J. Chowning, "Stria", 1977, from Turenas · Stria · Phoné · Sabelithe, LP. Digital Music Digital, 1988.

[27] E. Radigue, "Triptych, pt. 1", 1978, in Triptych, LP. Important Records, 2015.

[28] P. Schaeffer, "Acousmatics", 1966, in C. Cox, and D. Warner (Eds.), Audio Culture: Readings In Modern Music. Continuum International Publishing Group, 2004, pp. 76-81.

BEATINGS: A WEB APPLICATION TO FOSTER THE RENAISSANCE OF THE ART OF MUSICAL TEMPERAMENTS

Rui Penha
INESC TEC and Faculty of Engineering, University of Porto
rui.penha@inesctec.pt

Gilberto Bernardes
INESC TEC
gba@inesctec.pt

ABSTRACT

In this article we present *beatings*, a web application for the exploration of tuning and temperaments which pays particular attention to auditory phenomena resulting from the interaction of the spectral components of a sound, and in particular to the pitch fusion and the amplitude modulations occurring between the spectral peaks a critical bandwidth apart. By providing a simple, yet effective, visualization of the temporal evolution of this auditory phenomena we aim to foster new research in the pursuit of perceptually grounded principles explaining Western tonal harmonic syntax, as well as provide a tool for musical practice and education, areas where the old art of musical tunings and temperaments, with the notable exception of early music studies, appears to have long been neglected in favour of the practical advantages of equal temperament.

1. INTRODUCTION

The history of tunings and temperaments in Western music is closely related to the evolution of compositional practice, from Pythagorean tuning and its perfect consonances of octaves, fifths and fourths to equal temperament and the spread of chromaticism. From its peak development phase in the common-practice period, tonal music has evolved in numerous ways to become a particularly successful syntax – one that still forms the basis for musical training in Western culture –, forming a plethora of tonal composition idioms such as today's pop and jazz music. Harmony is a primary, well-research element of tonal syntax, to which many theories have been devoted. These explain the principles regulating the tonal music syntax, by abstracting archetypical structures and rules from large corpora of Western tonal music [1, 1–6]. Each of these theories emphasizes different aspects that regulate harmony, including voice leading [2, 5], root progression [3, 4], and tonal tension [6] in an axiomatic and formalized manner.

More recently, psychoacoustic and cognitive studies have shown that the aesthetic origin of Western tonal harmony syntax appears to be consistent with perceptual auditory streaming principles [7–9]. Parncutt [7] presented a comprehensive theory of musical harmony explained by psychoacoustic and pitch-related elements of music perception. Huron [8] was particularly successful in outlining most axiomatic principles within Western tonal harmony theories from a perceptual viewpoint. These new findings not only prompt a new understanding of the auditory mechanisms regulating harmony, but can also promote new compositional strategies.

Motivated by the possibility to study auditory phenomena and draw a research agenda on topics across cognitive psychology and music theory, we present *beatings*, a web application for the real-time visualization of amplitude modulations and fusion created by the interaction of spectral peaks. These sensory phenomena are known to have an impact on the perception of consonance and dissonance [7,10,11] and, to the best of our knowledge, no other application allows the visual exploration of these phenomena in an explicit way.

Many potential applications exist for our work. First, it allows the performance of music using different tunings and temperaments readily from the browser, surpassing the need for an (historical) instrument which is able to cope with adjustable tuning. As an ear training tool, for the music student and/or professional tuner, *beatings* can offer a refined level of control over the several components tones of harmonic intervals. Furthermore, it offers a simple way to perform with different tuning systems, while analysing the interaction between spectral harmonic peaks in an accurate way. Finally, *beatings* constitutes a platform for future research in tuning, temperaments, and the acoustics of musical scales and harmony, in particular by unveiling physical correlates of sound. Untimely, by highlighting the direction of spectral peaks and interaction over time, we hope that *beatings* may shed some light on a limitation of Huron's [8] theory, which lacks an explanation for the sense of direction ("leading") that attends musical pitch successions in tonal Western music. As it stands, Huron's [8] perceptually-derived voice leading rules are equally effective in both directions, which clearly does not capture the essence of harmony writing in cases where contingent resolutions are required such as embellishments (e.g., suspensions, appoggiaturas) or the common voice leading of the third and seventh of dominant chords.

The remainder of this article is structured as follows. Section 2 reviews psychoacoustic phenomena of pure and complex tone interactions within one critical bandwidth as the basis of the mechanics and design principles of our web application, whose implementation we detail in Section 3. Section 4 presents an extensive plan for future work per application area, shedding some light on the possible uses

of the application as well as phenomena and problems it might allow us to observe and equate in greater detail. Section 5 concludes the paper and summarizes our contribution.

2. CRITICAL BANDWIDTH: FROM FUSION TO BEATINGS TO ROUGHNESS TO SMOOTHNESS

The critical bandwidth is a term coined by Fletcher [12] to describe the frequency bands of the human 'auditory filters'. These bands regulate innate auditory phenomena, including the tone sensations evoked by the superposition of frequencies, relevant to our paper.

Considering two frequencies f_1 and f_2 in Hz, instantiated at the same frequency and gradually displaced by increasing f_2, we now describe two important physical correlates of sound that occur within a critical band: frequency discrimination and beatings.

First, a single tone or unison is perceived when the frequencies are identical up to the limit of frequency discrimination, when two tones start to be perceived. The human ear tends to fuse the two tones up to a difference of between a half- and whole-tone (for the pitch register of musical instruments) depending strongly on the critical band and the individual [13]. Stumpf [14] developed this phenomena by drawing a theory of tonal fusion which explains "the tendency for some concurrent sound combinations to cohere into a single sound image" [8]. In [15], Huron showed that in the polyphonic writing of J.S. Bach, intervals that promote tonal fusion are avoided in favour of discernible lines, or clear independence across voices.

Beatings are an auditory phenomena created by the phase interaction (i.e., reinforcement and cancellation) between tones within a critical band. Perceptually, it results in a variation of volume (i.e., amplitude modulation) whose rate, or frequency f_b in Hz, can be calculated as the difference between the two frequencies, such that:

$$f_b = |f_2 - f_1| \tag{1}$$

When f_b is approximately 10 Hz we perceive 'slow' beatings, whose modulation rate can be easily followed by the ear. Musically, this effect is referred to as tremolo [16]. When the beatings' frequency increases to around 20-30 Hz, the tremolo sensation ceases to be heard and, instead, 'fast' beatings create roughness – an unpleasant sensation perceived up to one critical bandwidth. Above this limit, the roughness sensation between two pure tones is replaced by a smoothness sensation. Figure 1 summarizes the different perceptual phenomena resulting from the interaction of two frequencies in terms of identifying the transitions between the auditory phenomena mentioned above. Although Figure 1 establishes exact frequencies as limits for several auditory phenomena, these are rather mean values extrapolated from listening experiments, which may vary between individuals.

Partials of complex tones are also known to produce a beating sensation when they are a critical bandwidth apart, thus following the same principles detailed for two frequencies. As a result, the timbre of complex tones can affect our experience of several perceptual phenomena dictated by the critical bands. Fast beats, also know as roughness, have been recognized to be one of the most relevant aspects of innate influences on the perception of (sensory) dissonance, and thus affect our subjective experience of musical and harmonic dissonance [11, 13, 17].

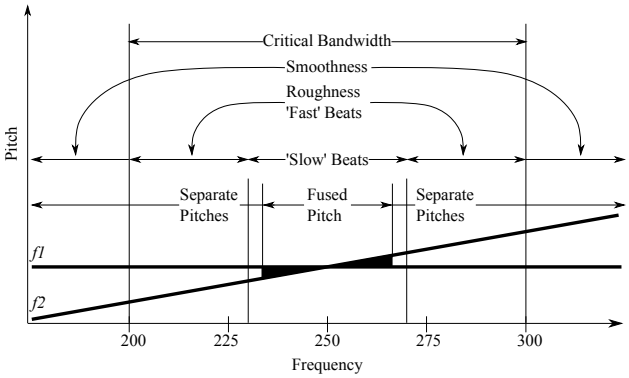


Figure 1. Pure tones interaction in terms of their perceptual fusion and beating qualities (adapted from [13]).

3. APPLICATION

beatings currently exists as a web application, [1] developed using p5.js [18], and is compatible with any desktop browser that supports the Web AUDIO API [19]. However, it is only fully functional in browsers that also support the Web MIDI API [2] [20]. In this section, we will detail its implementation and describe the most relevant user interface design decisions.

3.1 User Interface

The user interface of *beatings* (as seen in Figure 2) takes the form of circle divided in twelve parts that represent the division of the octave into notes, as established by the currently selected temperament. This representation was inspired by [11] and always includes the division of the octave in twelve equal parts (i.e., equal temperament) in a lighter tone for comparison. The detuning from equal temperament in cents is shown, also in a lighter tone, around the correspondent note name. Inside this main circle, a spiral represents approximately 8 octaves, with one turn per octave from A0 to C9, as inspired by the recent "The Snail" plug-in by IrcamLab [21].

To add a notes, the user can click on any note name to activate or deactivate it. To change the octave of an activated note, the user can click on one of the intersections between the note radius and the spiral and drag it up or down to change the octave. The lowest of the selected notes is represented by an arrow pointing to the centre of the circle and all of the currently selected notes are represented in a musical score visualization on the bottom left corner of the interface. [3] The number of harmonic partials shown for

[1] available at http://ruipenha.pt/beatings/.

[2] At the time of writing, only Chrome and Opera have Web MIDI support.

[3] All note representations can toggle between the use of sharps or flats by pressing the **f** key.

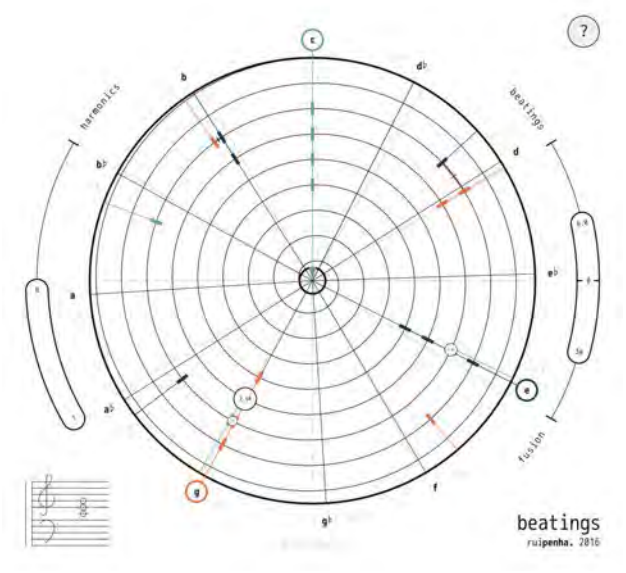


Figure 2. The graphical user interface of *beatings* showing a C major chord in the Werkmeister III temperament.

each selected note can be set using the interface element to the left of the main circle. To change the tuning of the last selected note, the user can use the **up** and **down** keyboard arrow keys to change it one cent [4] (or 0.1 cents, by holding **shift**) up or down, respectively.

By pressing and holding the **spacebar**, the user can listen to the currently selected notes. Each harmonic partial n shown is synthesized by a sinusoid of amplitude a

$$a = \frac{1}{n} \tag{2}$$

thus approximating a sawtooth wave.

Slow beatings are visualized by a circle that is centred at the middle frequency between the interfering partials. This circle expands and contracts at the frequency of the slow beating, also shown inside the circle, with its maximum and minimum amplitudes being proportional to the maximum and minimum amplitudes of the slow beating. This visualization aims to stimulate the auditory perception of the slow beating via multimodal perception. The criteria for this visualization can be set by adjusting the maximum frequency of the slow beatings shown by using the interface element at the top right of the main circle.

The fusion of partials is visualized by a grey arch that connects the interfering partials, with the middle frequency between those represented by a small slash. The criteria for this visualization can be set by adjusting the maximum interval in cents, using the interface element to the bottom right of the main circle. It is important to note that the criteria for both partial interference visualizations are set by the user and thus do not precisely represent the psychoacoustic limits as established by the relevant literature, even if these limits were taken into account when defining the respective range of the interface elements.

[4] I.e., one cent of a semitone, corresponding to an equal division of the octave in 1200 cents.

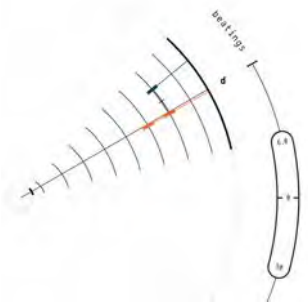


Figure 3. Detail of the graphical user interface of *beatings*, showing the isolation mode.

key	tuning / temperament
0	Equal Temperament
1	Pythagorean
2	Pietro Aaron (meantone)
3	Werkmeister III
4	Thomas Young
5	Just Intonation 1
6	Just Intonation 2
7	Just Intonation 3

Table 1. Default tunings and temperaments included in *beatings*.

To facilitate the auditory perception of these partial interferences, the user can press the key **i** to enter the isolation mode. This mode enables the isolation, by moving the mouse around the circle, of a particular part of the main circle for sound synthesis (as shown in Figure 3).

Some historical tunings and temperaments have been included in *beatings* and are accessible via the number keys, as seen in Table 1. While the selection of these tunings and temperaments is not exhaustive, it takes into account the history of European musical temperaments [11,22–24] and presents the main representative of pre-renaissance tunings (Pythagorean), one representative of renaissance meantone tunings (Pietro Aaron), one representative of baroque tunings (Werkmeister III), one representative of classical tunings (Thomas Young) and the ubiquitous equal temperament and just intonation. Just intonation is represented in three versions, all referring to C, in which the notes corresponding to D, F sharp / G flat and A sharp / B flat are tuned to, respectively: 1) minor tone, augmented fourth and harmonic minor seventh; 2) major tone, diminished fifth and grave minor seventh; 3) major tone, diminished fifth and minor seventh.

Finally, it is possible to export the current visualization as a PDF, by pressing the **s** key. As it is impracticable to convey the slow beatings in the PDF file using the same strategy as in the web application, these are represented by indexes inside the main circle (as seen in Figure 5). These indexes refer to the detailed descriptions that appear in a table to the right of the main circle, sorted in descending order from the maximum amplitude.

3.2 Web MIDI API

It was decided early on to use the Web MIDI API [20] in *beatings*, even if this standard is still in its early stages of deployment and, at the time of this writing, available only on a limited number of web browsers. This API provides easy access to the MIDI interfaces connected to the machine running the browser, allowing the use of MIDI controllers as an additional note input strategy. This permits the real-time playing of music in different temperaments or the rendering of MIDI files, using an inter-application MIDI router, such as the IAC Bus (on Mac OS X).

4. PROPOSED RESEARCH DIRECTIONS

4.1 Application improvements

We intend to continue the development of *beatings* by adding additional features to improve its merits as both a musical instrument and a research tool. The application currently lacks a way to save and retrieve temperaments and, consequently, a convenient way to share the users' proposals. Also currently missing is an easy way to embed playable custom temperaments or interval examples in web pages, which could be used to facilitate the dissemination of examples. A built-in MIDI player could facilitate the experience of the effect of different temperaments in different musical pieces. Finally, providing additional controls over the synthesis parameters (such as the global envelope or the amplitude and tuning of individual partials) would enhance not only the usefulness of *beatings* as a musical instrument, but also its ability to provide a more meaningful experience of the relation between timbre, tuning and the perception of consonance and dissonance [11].

4.2 Research in voice leading

As previously mentioned, our main motivation for the development of *beatings* was to enable further research in voice leading within tonal harmony, something we intend to pursue in the near future. The relationship between the history of tunings and temperaments and the evolution of harmony in Western musical culture is to be expected and has some striking coincidences, such as the apparent contiguity between the arising of meantone temperaments, that began altering the perfect fifths of Pythagorean tuning to favour consonance in thirds, and the transition of the fourth from the status of a perfect interval to the status of a dissonance, to be resolved downwards towards a third. If we observe a fourth between, e.g., C4-F4 in *beatings* using Pythagorean tuning (Figure 4) and using Pietro Aaron's meantone (Figure 5), we can easily see (and hear) that the stable fourth of the former is, in the latter, rendered unstable by the prominent slow beating of approximately 3.24 Hz between the fourth partial of C4 and the third partial of F4. The shortest path to resolve this instability is to descend the F4 to E4, that in Pietro Aaron's meantone temperament corresponds to the just intonation of the major third of C4 and is thus particularly consonant. Can we find more examples such as this one? Can the interrelation of slow beatings, fusion and the sensation of musical scale help to create the urge to lead specific voices up or down to solve particular dissonances?

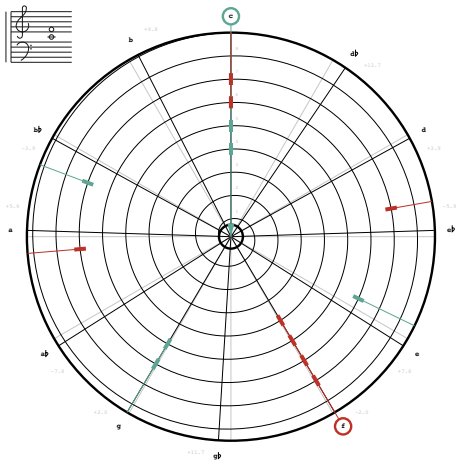


Figure 4. C4-F4 interval (perfect fourth) in Pythagorean tuning, shown with 8 harmonic partials for each note.

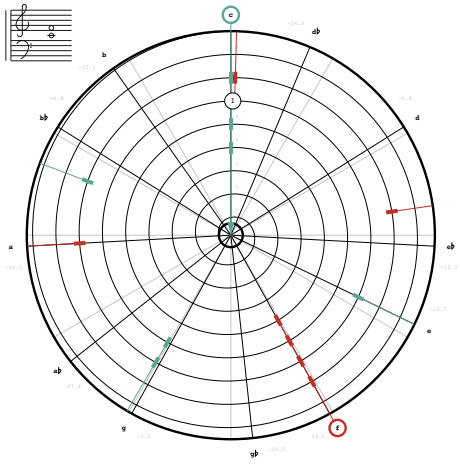


Figure 5. C4-F4 interval (perfect fourth) in Pietro Aaron's meantone temperament, shown with 8 harmonic partials for each note. The slow beating with label (1) has a frequency of approximately 3.24 Hz.

4.3 Beyond the Twelve-Tone Division of the Octave

An obvious limitation of *beatings* is the exclusive reliance on the twelve-tone division of the octave. This decision is related not only to the prevalence of this division in Western musical culture, but also because of the reliance on the MIDI protocol and the ubiquity of the keyboard as a MIDI interface. We would like, however, to include the capability of exploring tunings and temperaments with different divisions of the octave, acknowledging the contribution of other musical cultures to the art of tuning and temperament, as well as the inspiring work of Western composers such as Harry Partch.

5. SUMMARY

In this paper we have presented *beatings* and some of our motivations for its development. We believe that this application provides a novel and compelling way of exploring tunings and temperaments. The straightforward availability via web browser, along with the real-time synthesis and MIDI control capabilities, might help contemporary musicians to get acquainted with the effect of different tunings and temperaments, adjust them and choose before actually retuning an acoustic instrument (e.g., a harpsichord) or before searching for the same temperament in non-fixed tuning instruments (e.g., voice or bowed string instruments). In acoustics and psychoacoustics classes, the multimodal experience of slow beatings and the possibility to easily isolate fusion phenomena might prove helpful for students to explore and better understand the acoustical properties of harmony within different musical tunings and temperaments. Finally, we hope to contribute to the renewal of the interest in tunings and temperaments and, in particular, to the research of their impact on harmony and voice leading.

Acknowledgments

Project "TEC4Growth - Pervasive Intelligence, Enhancers and Proofs of Concept with Industrial Impact/NORTE-01-0145-FEDER-00002" is financed by the North Portugal Regional Operational Programme (NORTE 2020), under the PORTUGAL 2020 Partnership Agreement, and through the European Regional Development Fund (ERDF). This research is also supported by the Portuguese Foundation for Science and Technology under the post-doctoral grant SFRH/BPD/109457/2015.

6. REFERENCES

[1] F.-J. Fétis, *Trait Complet de la Thorie et de la Pratique de l'Harmonie*. Brandus et Cie., 1853.

[2] H. Riemann, *Vereinfachte Harmonielehre die Lehre von den Tonalen Funktionen der Akkorde*. Audener, 1983.

[3] J. P. Rameau, *Treatise on Harmony*. Dover, 1971.

[4] W. Piston, *Harmony*. Norton, 1978.

[5] H. Schenker, *Harmony*. University of Chicago Press, 1980.

[6] F. Lerdahl and R. Jackendoff, *A Generative Theory of Tonal Music*. The MIT Press, 1983.

[7] R. Parncutt, *Harmony: A Psychoacoustical Approach*. Springer, 1983.

[8] D. Huron, "Tone and voice: A derivation of the rules of voice-leading from perceptual principles," in *Music Perception*, vol. 19, no. 1, 2001, pp. 1–64.

[9] J. J. Bharucha, "Anchoring effects in music: The resolution of dissonance." in *Cognitive Psychology*, vol. 16, 1984, pp. 485–518.

[10] R. Plomp and W. Levelt, "Tonal consonance and critical bandwidth," in *Journal of the Acoustical Society of America*, vol. 38, 1965, pp. 548–560.

[11] W. A. Sethares, *Tuning, Timbre, Spectrum, Scale*. Springer, 2005.

[12] H. Fletcher, "Auditory patterns," in *Reviews of Modern Physics*, vol. 12, 1940, pp. 47–65.

[13] J. G. Roederer, *The Physics and Psychophysics of Music: An Introduction*. Springer, 1995.

[14] C. Stumpf, *Tonpsychologie (vol. 2)*. S. Hirzel, 1890.

[15] D. Huron, "Tonal consonance versus tonal fusion in polyphonic sonorities," in *Music Perception*, vol. 9, no. 2, 1991, pp. 135–154.

[16] G. Loy, *Musimathics: The Mathematical Foundations of Music*. The MIT Press, 2011.

[17] N. Iyer, B. Aarden, E. Hoglund, and D. Huron, "Effect of intensity on sensory dissonance," in *Journal of the Acoustical Society of America*, vol. 106, no. 4, 1999, pp. 2208–2209.

[18] L. McCarthy. (2015) p5.js. [Online]. Available: https://p5js.org/

[19] P. Adenot, C. Wilson, and C. Rogers. (2015) Web audio api, w3c working draft. [Online]. Available: https://www.w3.org/TR/webaudio/

[20] C. Wilson, Google, and J. Kalliokoski. (2015) Web midi api, w3c working draft. [Online]. Available: https://www.w3.org/TR/webmidi/

[21] IrcamLab. (2016) The snail. [Online]. Available: http://www.ircamlab.com/products/p2242-The-Snail/

[22] J. M. Barbour, *Tuning and Temperament: A Historical Survey*. East Lansing, 1951.

[23] S. Isacoff, *Temperament: How Music Became a Battleground for the Great Minds of Western Civilization*. Alfred A. Knopf, 2001.

[24] R. W. Duffin, *How Equal Temperament Ruined Harmony (and Why You Should Care)*. W. W. Norton and Company, 2008.

EXPLORING GESTURALITY IN MUSIC PERFORMANCE

Jan C. Schacher
Institute for Computer Music and Sound Technology ICST
Zurich University of the Arts
jan.schacher@zhdk.c

Daniel Bisig
Institute for Computer Music and Sound Technology ICST
Zurich University of the Arts
daniel.bisig@zhdk.ch

Patrick Neff
University of Zurich
Department of Psychology
patrick.neff@uhz.ch

ABSTRACT

Perception of gesturality in music performance is a multi-modal phenomenon and is carried by the differentiation of salient features in movement as well as sound. In a mix of quantitative and qualitative methods we collect sound and motion data, Laban effort qualifiers, and in a survey with selected participants subjective ratings and categorisations. The analysis aims at uncovering correspondences in the multi-modal information, using comparative processes to find similarity/differences in movement, sound as well categorical data. The resulting insights aim primarily at developing tools for automated gestural analysis that can be used both for musical research and to control interactive systems in live performance.

1. INTRODUCTION

By taking instrumental performance of a canonical contemporary piece as study object, this investigation aims at understanding which elements contribute to the multi-modal nature of music perception and how these elements are interrelated.

This investigation is carried out in the context of a larger research project which aims to develop analytical methods for the identification of gestures in composition and performance. The project complements a strong focus on artistic practice with a cross-disciplinary approach that integrates three academic disciplines. Psychological research explores gesture categories that inform music-perception. Music Technology uses motion data to recognise and categorise gestures in an automated way. Music Analysis builds a framework for gestures classification in composition and performance.

In the context of the cross-disciplinary research project the necessary skills for the multi-methodological approach are present and in the dialogue, a perspective is developed that can bridge between academic investigations and artistic practice, in particular with the terminology and the tools that are refined throughout the process.

2. BACKGROUND

Designing research methods that bridge between objective, data-driven methods and subjective, perceptual reporting is a standard approach in social sciences, but much less so in music research. Since music is a social, cultural as much as as physical phenomenon, blending the two perspectives [1] provides an overlapping field, that potentially describes the impact and import of music in a more appropriate manner, than using exclusively either one of the methods.

By triangulating between the four positions of the theories used to structure the work-flow, the methods used to carry out the investigation, the type and nature of the data collected and the changing roles of the investigators, the validity of results increases [2] and produce the essential effects of convergence, inconsistency, and contradiction [3]. For a more in-depth overview on mixed-method research please refer to [4] and in relation to motion analysis in music refer to [5].

Music analysis methods in the domains of empirical research comprise the rich set of music information retrieval (MIR) methodologies, that originate from the need to search and identify music pieces and have a set of metrics and descriptors that are unique to a given piece [6–8]. Similarly, movement research dates back at least to the nineteenth century with the chrono-photographies by Muybridge [9], and has numeric tools for extracting significant features from, for example, motion capture data [10].

On the qualitative side, *movement analysis* is an established topic in dance-research [11, 11], but also robotics [12] and physiology and rehabilitation [13]. Here it is particularly interesting to observe that perceived movement qualities, i.e., what makes movement expressive for our perception, has been formalised and is now usable in mathematical models as well as descriptive analysis, as we will discuss further on.

An additional pole in our configuration is a *terminological* and categorisations investigation about *musical gesture* [14, 15]. This domain is informed as much by body-related dimensions, body-instrument relationships as by musical categories such as phrase, chunk, segment, or semantic unit [16]. The question of musical content and the size of units to investigate is directly related to the temporal frames that are perceived as a musical unit, albeit this highly dependent on stylistic and other contextual elements of the music investigated.

2.1 Modelling the Methods

In mixed-method research the question of the balance between the qualitative information and the quantitative data is critical. Even when deploying mainly data-driven methodologies, the decisions about what data to treat in

what way are to a certain extent subjective. In this study, we explore a mixed method work-flow that is cyclical, and fluctuates between purely quantitative data-driven analysis with mathematical methods and subjective, perceptual qualitative interpretation, based on reports and observations.

From a methodological point of view, this should not represent a problem, provided there is clear an unambiguous declaration of which element is situated in which domain. Therefore a step by step description and assignment in categories between the dimensions at hand can shed light on the validity of the methods and the extracted insights.

In this complex work-flow that straddles the divide between objective, data-driven measurements and subjective, perceptual and self-reported impressions, the relationships between the elements are not merely cause-and-effect driven, which makes understanding the interrelation between aspects more complex.

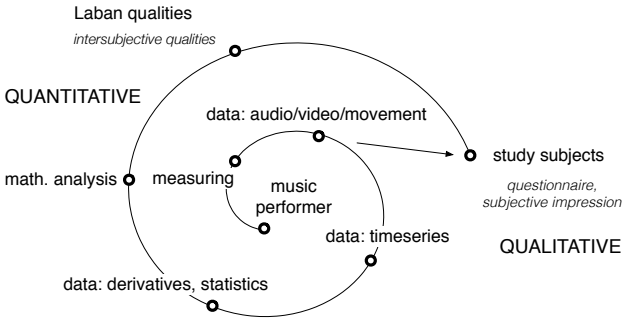


Figure 1: Cyclical model of mixed quantitative and qualitative methods. The axes stretch between qualitative and quantitative methods, and between tools and processes.

In an attempt to order and map the stages of the process and to assign the different outcomes and results to the main categories, the following cyclical map is proposed (see Fig. 1). Even if this arrangement neatly covers the steps in the work-flow, it is important to stress that this representation, as with each and every map, only represents one point of view on the configuration.

Reading the map is done by going from the centre to the periphery, from the musical performance, to measurement using technologies such as Motion Capture (MoCap) and audio video-recordings. Building on the collected data in time-series, a mathematical analysis leads to computable qualitative outcomes on the one hand and by evaluation through participants of a video of the performance to qualitative outcomes on the other hand.

3. THE CASE STUDY

A motion capture and audio-video recording session of a violoncello player performing the canonical piece 'Pression' by Helmut Lachenmann [17] was carried out (see Fig. 2). This composition is particularly interesting for a motion study on perceived effort, because the score prescribes the movements pertaining to specific playing technique rather than the sounding results. In addition, it focuses, as the name suggests, on extended playing techniques for the cello that have to do with rubbing, scratching and pressing in various ways with the hands and the bow on the strings and other parts of the cello.

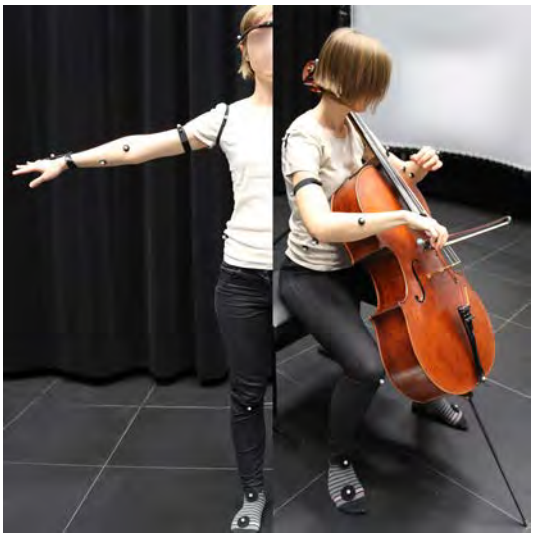


Figure 2: The Musician wearing passive reflective markers used for kinematic motion capture. Note the markers on the back of the hands (RFIN/LFIN), the tip of the bow (CLTP), the elbows (RELB/LELB), and the forehead (HDFR)

3.1 Multi-modal Approach

The multi-modal, blended nature of music perception that is occurring as much on the auditory, the visual as on the kinaesthetic sensory channels, means that investigation perceptual salience or 'gesturality' needs to occur in more than one modality in order to be meaningful. The modalities that are measurable from outside the performer are more practical to use than those relating to her body's physiological data, although they may be more telling about physicality and exertion during playing. In this study we use kinematic motion data from Motion Capture in combination with audio-data for the quantitative analysis, and for the qualitative investigation we use audio and video recordings evaluated by participants.

The point of departure for this investigation is perceived performance *effort* in a combined visual auditory case. This presupposes a definitions of effort. Apart from a physical and physiological measure, the term is used in motion analysis, in particular in the Laban Effort dimensions [11, p. 77]. Although in this system the term is used in an extended sense, it is still relevant for our purposes, since it addresses the human perception of effort, rather than just the measurable physical one. As we will see, the transfer from subjective to objective evaluation of these dimension also forms part of this investigation (see 3.3).

An additional core concept we focus on is that of *gesturality*. As a high-level concept that encompasses direct perception, semantic content and other psychological, affective factors, it serves to frame the more detailed systematisation found principally in the literature n musical gesture. After having mapped out the use of the term 'gesture' in this field the core terminology was selected and used to structure the responses of subjective impressions from participants (see 3.2).

An interesting parallel in this specific case can be drawn by comparing these gestural categories to musical ones in Lachenmann's 'Klangtypen der neuen Musik' [18] and Smalleys spectro-morphological Sound-shapes [19].

3.2 Qualitative Methods

Segments were preselected independently by all members of the research team from the entire piece. The primary criteria for the preselection were diversity and exemplariness of the segments regarding the piece. Five segments made it into the final selection and serve as materials for the mixed method workflow with quantitative music and movement analysis as well as qualitative third-person subjective ratings and categorisations from watching videos of the segments. Notably, for the qualitative ratings and categorisation the segments were condensed to a single gesture between three and six seconds in length.

In this survey, participants (n=26) were instructed to rate the segments based on their impressions (using various judgements of previous work [20] complemented by a genuine judgement of general gesturality) and furthermore categorise them into the concepts and terminologies presented in the previous section [14, 15, 21–23]. Within the framework of the study, we looked at the ratings of the general gesturality parameter and by introducing two main category systems of:

functional distinction between communicative, sound producing, sound-facilitating and -accompanying nature of gestures [15], between 'ergotic' [1], epistemic [2] and semiotic [3] gestures [14] as well as between gestures 'helping' the production of melody, harmony/musical structure, timbre, sound level, rhythm and tempo [22].

morphological distinction between trajectory-, force- and pattern-based primitives [23] as well as between impulsive, sustained and iterative morphologies [15]. Beyond that, participants of the survey were also invited to leave comments to their respective choices of categorisation. These comments serve as a verbalised pool of data flanking the categorisation and are indicative of reasoning about features of the video segment which led to the choice of categories.

The data sets gathered for each segment, i.e., single gesture, enable the evaluation and discussion of the quantitative continuous sound and motion data from the perspective of momentary qualitative data.

3.3 Quantitative Methods

The quantitative analysis of the musical performance is based on the extraction of lower and higher level features from multi-modal recordings that consist of synchronised audio and motion capture data. From the audio data, core features such as loudness (RMS measure), centroid, brightness, ad flux are extracted [8]. From the motion capture data, position time series are extracted for three markers placed on the forehead, back of the left hand and back of the right hand of the musician and a single marker placed on the tip of the bow. Prior to any feature extraction, the position time series are smoothed using a running average with a time window of ten samples. The computation of movement features is done by a software that forms part of the Machine Learning Workbench software tool chain [24]. For each individual marker, three kinematic features and three Laban effort features are calculated. The kinematic features comprise the first three temporal derivatives of the position time series (velocity, acceleration, jerk) and their absolute scalar values (speed, scalar acceleration, scalar jerk). These lower level features directly represent physical properties of body movement. The Laban effort features comprise weight effort, flow effort and time effort [11]. These higher level features compute from kinematic input data movement properties that are more closely related to qualitative aspects of movement such as dynamics, energy and expressiveness than the kinematic data themselves. Weight effort indicates the forcefulness of movement and discriminates between powerful and gentle movement qualities. Time effort reflects a sense of urgency and differentiates between quick and sustained movements. Flow effort represents the continuity of movement and distinguishes between free and bounded movements.

The implementation of the Laban feature extraction functionality is based on a recently published review of algorithms for calculating expressive motion descriptors [25]. For each Laban effort calculation, the input kinematic data are aggregated over a window size of ten samples. Following the feature extraction calculations, all audio and motion feature time series are normalised over the duration of the entire recording and subsequently truncated to the duration of each of the three performance segments. All time series corresponding to the same segment are then merged into a single file, whose content is rendered via a simple graphical plotting routine. This routine superimposes equivalent motion features for all marker positions and stacks different motion features and audio features from top to bottom. This visualisation forms the basis for a visual interpretation and comparison between quantitative, qualitative and audiovisual performance data.

4. DATA ANALYSIS

For sake of brevity and clarity we chose to focus on three segments. [4] In the following section, a brief and non-exhaustive quantitative and qualitative analysis of each segment is presented.

4.1 Segment Two

This segment is characterised by two short alternating scratches and plucking on the strings below the bridge at the edge of the string-holder.

Quantitative Assessment The bowing movements manifest in the quantitative data as clear correlations among peaks in the kinaesthetic and effort curves of the right hand and peaks in the sound loudness. The brevity

[1] "In the first function, ergotic, there is ... only energy communication between the hand and the object." [14]

[2] "epistemic, is typically performed by our capacity of touch and muscular/ articulatory sensitivity" [14]

[3] "The third function, semiotic, is that of meaning, of communicative intent. It's the gestural function per se." [14]

[4] See video of the entire piece and the individual segments: `http://mgm.zhdk.ch/?p=2021`

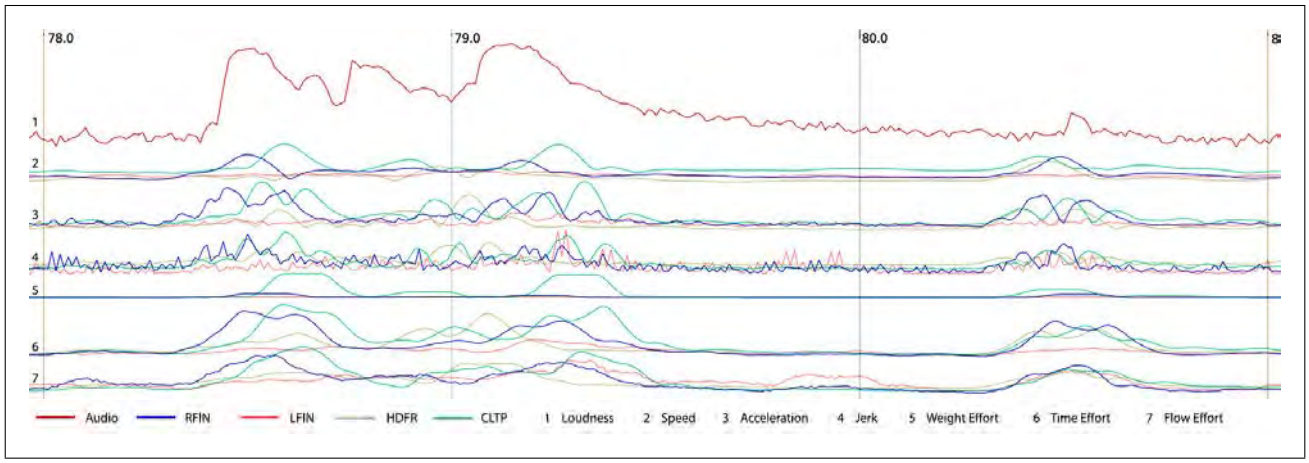


Figure 3: Continuous data plots of segment two.

of the velocity peaks in the right hand movement and their flanking by pronounced peaks in the acceleration and time effort curves are indicative of an impulsive bowing style. During bowing, peaks in the movement of the right hand and sound loudness are synchronised whereas peaks in the movement of the bow are delayed. These delayed peaks correspond to a pronounced removal of the bow after bowing. This removal exhibit a strong weight effort, time effort and flow effort, which is indicative of a forceful playing style.

Qualitative Assessment First, the survey showed a clear agreement between the participants categorising this segment as sound producing (21 of 26 participants) [15]. In the functional category system of Cadoz [14] this segment was rated ergotic in nature by the majority of participants (17/26). Looking at detailed aspects of sound production, 10 participants had the impression that the movements in this segment mainly helped the sound level whereas 7 participants chose rhythm and 5 timbre. The distribution of answers in this categorisation system [22] is therefore not showing conclusive agreement. The morphological aspects of movements [15] in this segment were clearly rated as impulsive by a large majority (23/26). Finally, concerning gestural primitives [23] participants agreed on the dominance of force-based primitives (22/26).

Regarding the rating of overall gesturality (on a scale of 1-5 with 1 being low and 5 being high in gesturality), this segment scored the lowest with a mean of 2.65.

4.2 Segment Three

In this segment, the end of the pig-sty scratching leads to the right hand slowing bowing on the bridge's face under the strings and the left hand cyclically rubbing and hitting on the fingerboard and the body of the cello.

Quantitative Assessment Percussive movements that occur when the left hand or the bow hit the instrument are characterised by a clear and strong synchronisation between peaks in movement features and peaks in sound loudness. These forceful and impulsive movements manifest as synchronisations that are visible across all movement features. The last loudness peak shows a clear correlation with the rubbing movements of the left hand but is de-correlated with the bowing movements of the right and bow tip. This deviation in feature synchronisation is a good indicator for the degree of coordination among different sound producing movements. During the rubbing movements of the left hand, the speed, acceleration, jerk and time effort curves show a repetitive pattern of pronounced peaks whose frequency and amplitude gradually increases. It is during these rubbing movements, that the movement of the forehead exhibits an interesting transition. During less effortful left hand movements, the forehead movement features are synchronised with the movement features of the right hand. As the left hand movements increase in effort, the features in the forehead movements become synchronised with the left hand movements instead. Accordingly, the amount of correlation between sound producing and non-sound producing movements can serve as an indicator for the amount of emphasis that is being put into a sound producing movement.

Qualitative Assessment As in the last segment, a clear agreement between participants was observable as this segment was categorised sound producing (21 of 26 participants). Concerning further functional categorisation, the segment was rated ergotic in nature by 13 participants whereas 8 participants chose the category 'epistemic' leaving 5 participants with a semiotic categorisation. The sound helping categorisation system again produced less agreement on a single category than other systems: 10 participants chose rhythm, 6 timbre, 4 harmony/musical structure, 3 sound level, 2 tempo and 1 'other'. Similar to that, the agreement in categorisation of morphological features was not clear-cut with half of the participants choosing impulsive (13), 7 iterative, 2 sustained and a total of 4 participants 'other'. Lastly, the absence of clear agreement continues in the categorisation of the gestural primitives with 10 participants opting for pattern-based primitives, 9 force-based, 5 trajectory-based and 2 participants 'other'.

The rating of gesturality produced an average of 4.04 with a large agreement amongst the raters rendering this segment the most gestural of all three.

4.3 Segment Four

In this segment the right hand plays a *saltando*-technique with the bow placed under the bridge. The left hand is passive and muting the strings.

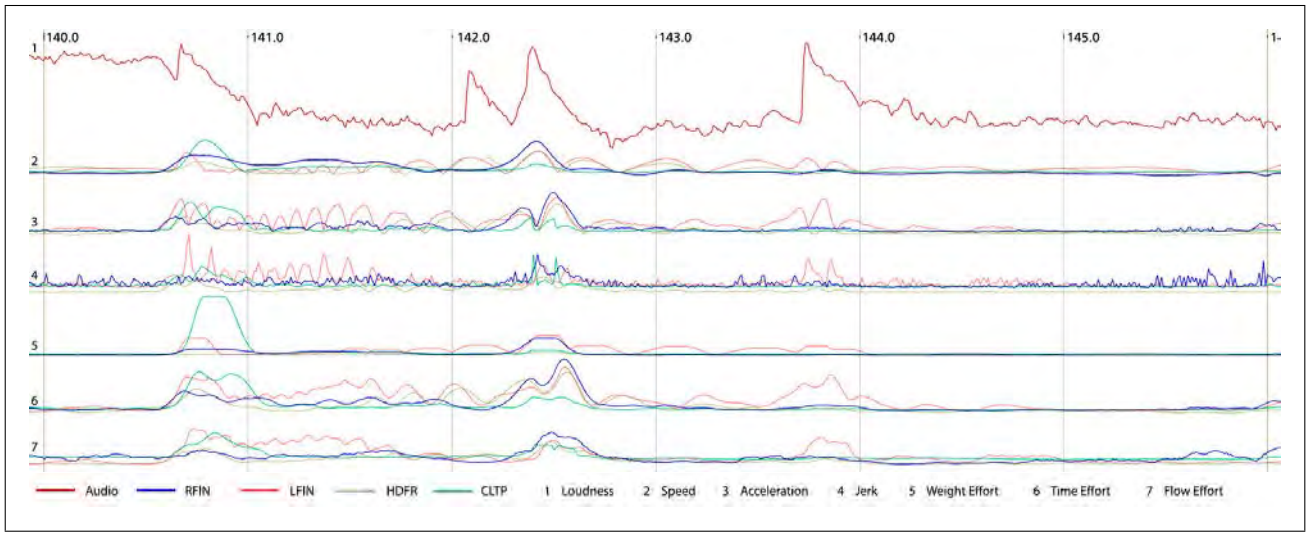


Figure 4: Continuous data plots of segment three.

Quantitative Assessment The percussive motions of the bow show repetitive peaks in jerk and elevated values in time effort and flow effort. Repetitive patterns in these motions are either actively actuated or result from a passive bouncing of the bow. These two types can be clearly distinguished based on the shape and correlation between sound loudness and the jerk and flow effort of the bow tip and right hand. For active repetitions, the features related to sound loudness and bow tip motion show constant amplitudes whereas for passive repetitions, the features decay in amplitude. Similarly, for active repetitions, the features of the right hand show high values. For passive repetitions, the feature values are low. In two cases, the head movement shows a peak in speed which precedes a percussive bow movement. These peaks operate as signifiers for the upcoming sound producing movements.

Qualitative Assessment In the last of the analysed segments, the agreement on a sound producing nature of the segment was not distinct with only 14 participants choosing this option. All other options were chosen to an equal extent with 4 participants at each of the categories of communicative, sound-facilitating and sound-accompanying. In the functional categorisation system, again 'ergotic' was favourably chosen by 12 participants whereas epistemic and semiotic scored with 6 and 8 choices respectively. Regarding the detailed aspects of sound production, 15 participants agreed on rating the movements helping the rhythm with no other category scoring more than 4. A similar conclusive agreement is observable in the morphological categorisation as the majority (17 participants) clearly rated the movements in the segment as being iterative. Finally, the overall rather distinct categorisation of this segment is also evident in the gestural primitives with 18 participants rating the segment as pattern-based.

Assessing the perceived general gesturality the segment was rated gestural with a mean of 2.92. This marks the segment as being the second most gestural of the 3 selected segments.

5. DISCUSSION

The following discussion starts with an evaluation of the similarities and differences between the quantitative and qualitative analysis results. We will argue that some of the differences result from the specifics of the experimental setup and propose means to modify and extend the method. From this, we will try to deduce general principles for employing quantitative and qualitative analysis in complementary ways.

Segment Two There exists a good correspondence between the quantitative data that indicate a impulsive and forceful playing style and the qualitative rating in the morphological categories.

On the other hand, the agreement among participants is not clear when it comes to the sound helping categorisation. For instance it is difficult to discuss why a majority (10/26) perceived the movements as primarily helping the sound level. In looking at Fig. 3, it is at least obvious that the motion and Laban derivatives accumulate in amplitude under the increased sound level. This is contrasted by respective sections of low movement and sound levels delineating an interaction of movement effort and related changes in sound level. On the other hand, comments like:

"Rhythm is the most pervasive feature of this short sequence. Of my memories of the seen, rhythm is the most important" [5] underline salient features of the alternative choices (here: helping the rhythm). The theorised factors for the choice of the sound helping category furthermore ease the understanding of the large agreement in the morphological categorisations of impulsiveness and force-based primitives. Impulsiveness is most probably perceived by the synchronised and energetic movements of the head and the rest of the body. In these moments the force applied emerges and becomes tangible which may be related to peaks in the acceleration data of these compact movements.

[5] "Der Rhythmus ist die aufdringlichste Komponente dieser kurzen Sequenz, in meiner Erinnerung an das Gesehene hat der Rhythmus die grösste Wichtigkeit"

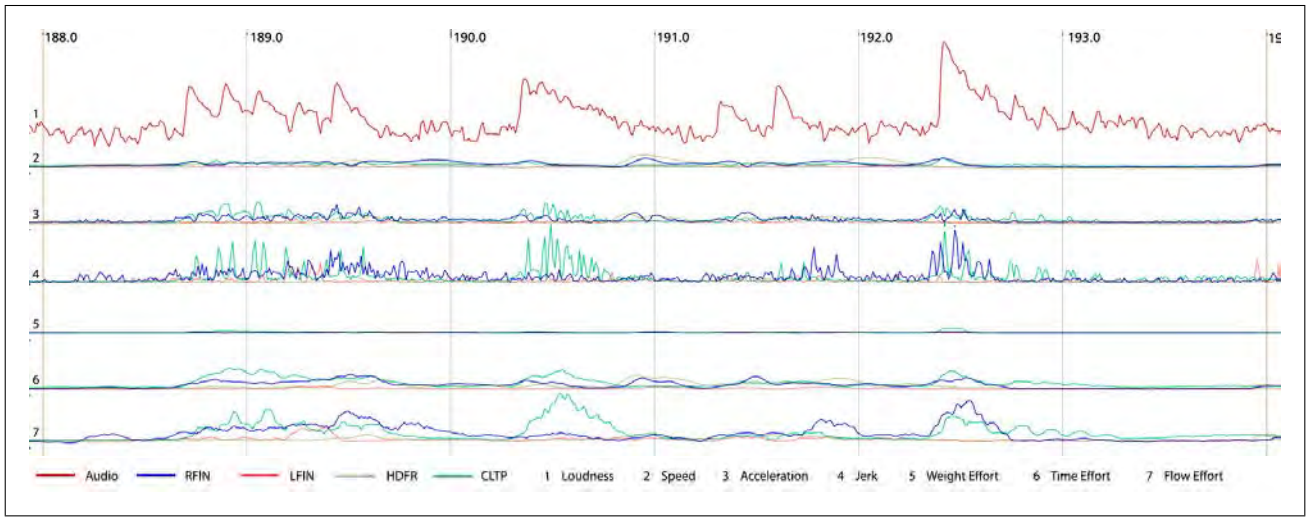


Figure 5: Continuous data plots of segment four.

Segment Three This segment is characterised by a large variety of different playing styles which causes a partial disagreement in the qualitative evaluation concerning the functional and morphological categories. Only a small majority of the participants agreed on a predominant presence of impulsiveness. With 7 choices of the category iterativity and 3 participants opting for the 4th open categorisation entitling it with "impulsive and iterative" and alike, the segment is not clearly categorised. This overall impression of broadly distributed categorisation data rather than clear-cut agreement of this segment is also impressively discernible in the equal distribution between pattern- and force-based primitives. The following comments may shed some light on the difficulties in decision: "The rhythm of the gesture and accompanying sound made me choose the pattern-based option. The sound rhythms translate well into patterns." "The beat implicates a sudden change. It is repeated but still very force-based."

A similar amount of disagreement exists in the functional categorisation. This might be caused by the gradual changes in the iterative left hand movements that convey an epistemic rather than ergotic function and the variability of the correlations between head and left hand movements which indicate a semiotic rather than ergotic categorisation. These observations may be confirmed by statements like the following annotating the epistemic choice: "The performer seemed to be experiencing the tactile structure of the different cello parts directly in her gestures. She almost seemed to be experimenting with how different sounds can be produced by different materials."

Segment Four There is a clear agreement between subjective and quantitative analysis concerning the rhythmical and pattern-based characteristics of the sound producing movements. Compared to the morphological categorisation of impulsiveness in the two other segments, participants opted for the iterative nature and pattern-based primitives for this segment.

Less agreement was observable in the functional categorisations with, for example, half of the participants choosing sound producing. Some comments on communicative aspects of head movements and mimic of the eyes with statements may explain this partial disagreement: "As I wrote earlier, the musician seems intent on communicating a mood or feeling by using her head movements and eyes." The described head and eye movements could also explain the differences in the rating of the functional categories as 8 participants chose semiotic, which is possibly related to the audience-directed communicative aspects of these movements. Furthermore, with 6 participants opting for the epistemic category, these movements can also be interpreted as personal moments of discovery by the performer in the interaction with the instrument and the material. A respective comment emphasises this observation: "I think that the epistemic function is also present here but to a lesser degree: one could explore the strings by these rhythmic movements." [6]

5.1 Gesturality

Looking at the rating of general gesturality of the segments, segment three was rated the highest followed by segment four and two. The questions which arises at this point is about the salient features in the segments which led to the respective ratings. First, we theorise that the amount and variety of movement as well as sound seems directly correlated to the gesturality scores. This observation is certainly limited to a large degree by the length of the segment and to a lesser degree by the selection of the segments, which may explain the dominance of segment three. Second, we reason that the ratings may also be influenced either by (salient) features not captured in the sound and motion data (e.g. mimic, communicative aspects of upper body/torso movements) or by more distal features like personal dispositions of the participants or features inherent to the musical material with their socio-cultural embedded-ness.

5.2 Differences

A number of reasons for the existence of differences between the subjective and quantitative analysis of the performance may exist.

[6] "Hier denke ich, dass auch die epidemische Funktion dabei ist, jedoch zu einem geringeren Anteil. Denn durch die rhythmische Bewegung könnte man auch die Saiten besser kennenlernen ."

Dimensionality differences The data acquisition process reduces the performance characteristics to a few dimensions. As a result, the quantitative analysis lacks data that is relevant in the qualitative analysis, for example data related to the timbral and melodic qualities of the music or facial expressions. The difference in the type and amount of data available for quantitative and qualitative analysis could be alleviated by including more sophisticated audio analysis and by integrating facial or gaze-tracking.

Attention differences The quantitative analysis processes all data dimensions individually and concurrently and assigns equal relevance to each of them. In the subjective analysis, by contrast, each participant focuses his or her attention on the most salient features only and the analysis results in a overall evaluation of high- level characteristics of the performance. These differences can be partially bridged by combining multiple 'MoCap' marker positions before conducting the analysis and by weighting the influence of each marker according to some salience criteria.

Contextual differences For the quantitative analysis, the sequence of observations of the performance segments is irrelevant, since the corresponding algorithms don't show any memory effects. The qualitative analysis, however, is strongly influenced by the sequence of segments, since human subjects tend to pay particular attention to differences between the segments. For this reason the sequence of presented segments was randomising in the qualitative analysis. This difference could be reduced by integrating an attenuation factor and a gradually shifting baseline into the quantitative analysis.

Correlating Sound and Movement It is difficult to analyse sound timbre effects in relation to sound loudness and movement features. In the case of the present composition with its array of noisy extended playing techniques, the task of extracting the physical motion and effort from the sonic content is challenging. Additional sound dimensions such as spectral centroid, flux, noisiness etc. could to be taken into account. The non-standard playing techniques makes using the Laban Motion Dimensions and their computable descriptors difficult to use. These descriptors seem useful to evaluate full body movements in dance (i.e. many joints) and less for evaluating movements of individual body parts in instrument playing (i.e. single joints).

Quantitative versus Qualitative Evaluation Potentially important expressive aspects of the performance are not detectable from 'MoCap' data alone because:

- facial expression changes are indicators of expressivity but not acquired in 'MoCap' data.
- small head movements have strong visual impact (carrier of semantic information) but do not appear as significant peaks in 'MoCap' data.
- upper body movements (important sound facilitating movements) show little significant features in 'MoCap' data.

Large differences in absolute values of motion features among different markers, e.g., a bow tip motion and a head movement, do not correspond to relevance in subjective interpretation. A small head movement, for example, might be considered a more important sound accompanying or facilitating gesture than a large bow motion, because it carries different signification: the former is perceived as an expressive gesture, whereas the latter as a controlled instrumental action.

6. CONCLUSIONS AND OUTLOOK

The in-depth analysis and discussion on morphological as well functional characteristics, as well as the differential observation both between qualitative and quantitative methods, and within the a single analysis category, changed the focus of this project. This meant moving from a hypothesis based on 'effort' to the wider and more versatile concept of salient features. As the interpretation shows, quantitative and qualitative data can be put into relationship and their mutual complementarity can be shown. This means that effects and statement observed through the qualitative methods may be better understood thanks to the interpretation of data carried out with the quantitative methods. The principal remaining incompatibilities have to do with the semantic, expressive content on the level of the musical performance (see 5.2). The highest level of perception is also indicated in the participant's overall ratings on 'gesturality', which in this piece inherent to the musical material (see 5.1). A stronger correspondence between the data-driven analysis and the subjective impressions by participants could be achieved by adding facial measures. Nevertheless, in order to have more encompassing interpretation the complementarity of quantitative data and qualitative statement is useful, and does not need to be brought into total alignment.

Starting from the hypothesis that expressivity in musical performance is carried by the perception of salient features in movement, we come to the following insights: Perceived gesturality is depending on salient features in movement. In a quantitative measure these are salient contrast values, whereas in the subjective, qualitative domain they reside in aspects that are not by necessity those of the instrumental actions, but rather those containing communicative information, or in an epistemic sense emphasise the exploration of the instrument through touching and action.

The results of this enquiry indicate that it would be useful to include more dimensions for the methods used to achieve a finer differentiation through: different sensors (facial tracking, physiological sensor, muscle tension sensors, skin conductance, heart rate, brain responses etc.); extending questionnaire techniques (weighting/ranking instead of forced-choice; extended setups such as quantitative continuous measurements of audience response (e.g., by means of pressure sensors) as well as live ratings.

The multi-methodological approach conducted in this research is guided by a motivation that is strongly rooted in musical practice. This motivation is based on the desire to enable computer-based interactive systems to respond to higher level qualitative and expressive cues in a musical live performance. In such a setup, the normal expressivity of a human performer would not need to be constrained

by functional necessities for controlling a computer-based instrument. Rather, this expressivity becomes as an intrinsic carrier of meaning that is readable not only by the audience but also by the accompanying computer system. The identification of correspondences between a quantitative analysis of sensorial data and qualitative and subjective evaluations of audiovisual performance media serves as a very first step towards this goal. This step helps to inform the design and implementation of feature extraction and machine learning algorithms that are able to mimic human audiences in the recognition of expressivity in a real-time, live-performance stage context. In order to have these feature-recognition processes available in an interactive system, those quantitative measures that are as close as possible to qualitative subjective interpretation could prove to have a powerful effect on expressivity for interactive compositions in music and dance.

Acknowledgments

This investigation was carried out within the 'Motion Gesture Music' Project at the Institute for Computer Music and Sound Technology of the Zurich University of the Arts, and was funded by the Swiss National Science Foundation Grant No. 100016_149345.

7. REFERENCES

[1] G. Fauconnier and M. Turner, *The way we Think: Conceptual Blending and the Mind's Hidden Complexities*, 2008th ed. New York, USA: Basic Books, 2003.

[2] E. Webb and K. E. Weick, "Unobtrusive measures in organizational theory: A reminder," *Administrative Science Quarterly*, pp. 650–659, 1979.

[3] N. K. Denzin, *The research act: A theoretical introduction to research methods*. New York, USA: McGraw-Hill, 1978.

[4] R. B. Johnson, A. J. Onwuegbuzie, and L. A. Turner, "Toward a definition of mixed methods research," *Journal of mixed methods research*, vol. 1, no. 2, pp. 112–133, 2007.

[5] J. C. Schacher, "Moving Music – Exploring Movement-to-Sound Relationships," in *Proceedings of the International Symposium on Movement and Computing (MoCo'16)*, Thessaloniki, Greece, 5.–6. July 2016.

[6] M. Fingerhut, "Music information retrieval, or how to search for (and maybe find) music and do away with incipits," in *IAML-IASA Congress, Oslo*, 2004.

[7] J. S. Downie, "Music information retrieval," *Annual review of information science and technology*, vol. 37, no. 1, pp. 295–340, 2003.

[8] O. Lartillot and P. Toiviainen, "A Matlab Toolbox for Musical Feature Extraction from Audio," in *International Conference on Digital Audio Effects, DAFx*, 2007, pp. 237–244.

[9] E. Muybridge, *The human figure in motion*. Courier Corporation, 2012.

[10] B. Burger and P. Toiviainen, "MoCap Toolbox-A Matlab toolbox for computational analysis of movement data," in *Proceedings of the Sound and Music Computing Conference, SMC 2013*, Stockholm, Sweden, 2013.

[11] R. Laban, *The Mastery of Movement*, 4th ed. Alton, Hamphire, UK: Dance Books Ltd., 1950 (1980/2011).

[12] H. Knight and R. Simmons, "Layering laban effort features on robot task motions," in *Proceedings of the Tenth Annual ACM/IEEE International Conference on Human-Robot Interaction Extended Abstracts*. ACM, 2015, pp. 135–136.

[13] E. Davies, *Beyond dance: Laban's legacy of movement analysis*. Routledge, 2007.

[14] C. Cadoz, M. M. Wanderley *et al.*, "Gesture-Music," *Trends in Gestural Control of Music*, vol. 12, pp. 71–94, 2000.

[15] R. I. Godøy and M. Leman, *Musical Gestures: Sound Movement and Meaning*. New York, USA: Routledge, 2010.

[16] R. I. Godøy, A. R. Jensenius, and K. Nymoen, "Chunking in music by coarticulation," *Acta Acustica united with Acustica*, vol. 96, no. 4, pp. 690–700, 2010.

[17] T. Orning, "Pression – a performance study," *Music Performance Research*, vol. 5, pp. 12–31, 2012.

[18] H. Lachenmann, "Klangtypen der neuen Musik," in *Musik als existentielle Erfahrung*. Wiesbaden, Germany: Breitkopf & Härtel, 1966, pp. 1–20.

[19] D. Smalley, "Spectromorphology: explaining soundshapes," *Organised sound*, vol. 2, no. 2, pp. 107–126, 1997.

[20] B. W. Vines, C. L. Krumhansl, M. M. Wanderley, I. M. Dalca, and D. J. Levitin, "Music to my eyes: Cross-modal interactions in the perception of emotions in musical performance," *Cognition*, no. 118, pp. 157–170, 2011.

[21] A. Jensenius, M. Wanderley, R. Godøy, and M. Leman, "Musical Gestures, Concepts and Methods in Research," in *Musical Gestures, Sound, Movement and Meaning*, R.-I. Godøy and M. Leman, Eds. New York, USA: Routledge, 2010.

[22] I. Poggi, "Body and Mind in the Pianist's Performance," in *Proceedings of the 9th international conference on music perception and cognition (—CMPC9)*, M. Baroni, A. R. Addessi, R. Caterina, and M. Costa, Eds., Bologna, Italy, 2006, pp. 1044–1051.

[23] I. Choi, "Cognitive Engineering of Gestural Primitives for Multi-modal Interaction in a Virtual Environment," in *International Conference on Systems, Man, and Cybernetics*, vol. 2. IEEE, 1998, pp. 1101–1106.

[24] J. C. Schacher, C. Miyama, and D. Bisig, "Gestural electronic music using machine learning as generative device," in *Proceedings of the International Conference on New Interfaces for Musical Expression, NIME'15,*, Baton Rouge, USA, May 31 – June 3 2015.

[25] C. Larboulette and S. Gibet, *A Review of Computable Expressive Descriptors of Human Motion*. Vancouver, Canada.: Proceedings of the Second Workshop on Motion and Computing MOCO'15, August 14–15 2015.

AUTHORING SPATIAL MUSIC WITH SPATDIF VERSION 0.4

Jan C. Schacher
Institute for Computer Music and Sound Technology, ICST
Zurich University of the Arts
Zurich, Switzerland
jan.schacher@zhdk.ch

Nils Peters
Centre for Interdisciplinary Research in Music Media and Technology, CIRMMT
Montreal, Canada
nils.peters@mail.mcgill.ca

Trond Lossius
Bergen Center for Electronic Arts, BEK
Bergen, Norway
trond.lossius@bek.no

Chikashi Miyama
Institute for Music and Acoustics, ZKM
Karlsruhe, Germany
miyama@zkm.de

ABSTRACT

SpatDIF, the Spatial Sound Description Interchange Format is a light-weight, human-readable syntax for storing and transmitting spatial sound scenes, serving as an independent, cross-platform and host-independent solution for spatial sound composition. The recent update to version 0.4 of the specification introduces the ability to define and store continuous trajectories on the authoring layer in a human-readable way, as well as describing groups and source spreading. As a result, SpatDIF provides a new way to exchange higher level authoring data across authoring tools that help to preserve the artistic intent in spatial music.

1. INTRODUCTION

SpatDIF, the Spatial Sound Description Interchange Format [1, 2], is an initiative by musicians and researchers for the development of an industry-independent, light-weight, human-readable syntax for storing and transmitting spatial sound scenes. SpatDIF addresses the lack of an independent, cross-platform and host-independent solution for spatial sound composition. It is implementation-agnostic and not tied to a specific technical platform, programming language or file-format. Representing a high-level structure that embodies typical authoring and performance work-flows in spatial sound, it comprises a hierarchical syntax of descriptors, a set of basic definitions of units and coordinate dimensions, a number of methods and algorithms for spatial sound transformations, and comes with a set of best-practice examples in several markup-languages or network streaming protocols.

This results in a non-synchronous and potentially sparse description of spatial sound scenes, that is aimed at providing interoperability between different spatial sound rendering tools and spatial music venues. Additionally it serves as a storage format for archival purposes [3]

The SpatDIF syntax is implemented in a C/C++ software library that facilitates the integration in host environments [4], for example in MaxMSP and PureData as 3rd party externals, but also in other environments such as Supercollider [5] and Zirkonium [6], as well as Android mobile platforms [7].

While integrating SpatDIF into compositional environments and artistic workflows we realized that artistic intentions of spatial movements are not fully preserved without an ability to define and store continuous trajectory information. We decided to extend SpatDIF accordingly, because to our knowledge, such features are out of scope of other open formats tailored towards commercial or broadcast applications such as ITU-R BS.2076 [8].

1.1 The Stratified Approach

SpatDIF is a structured system for describing the different aspects of a spatial sound workflow where a number of different aspect come into play. An analysis of a variety of systems and tools, as well as similarly complex transmission systems, led to the grouping and organization of the different processes and tasks into a model that comprises many, but not all elements of such a work-flow.

In 2009 the authors of [9] proposed a layered model that describes the relationships and mediation processes between essential components in sound spatialization. This model comprises processing layers (see left column of Fig. 1) ranging from the low-level Physical Domain, which comprises devices that create the acoustical signals, such as loudspeakers, up to the highest-level description on the Authoring layer, which describes the organizing and dynamic processes that drive e.g., movement within the scene. SpatDIF is based on this model and with the latest iteration presented in this paper, SpatDIF defines descriptors from the processing layer two up to processing layer six.

The previously published SpatDIF version 0.3 [2, 10] describes the core elements of a sound scene as well as some of the lower level rendering and dispatching information. The information at that stage was oriented towards rendering of spatial sound content, carrying in a temporally quantized way the instructions necessary for the playback-system to trigger sound-files or route audio signals to sources within the scene and to subsequently position these source entities in the rendered sound scene.

SpatDIF version 0.3 enables the description of the appearance and disappearance of sound entities in the scene, the assignment of media content to the entities, and the evolution of the geometrical properties of those sound entities in the scene over time such as position changes. This discretized representation of a sound scene can be practi-

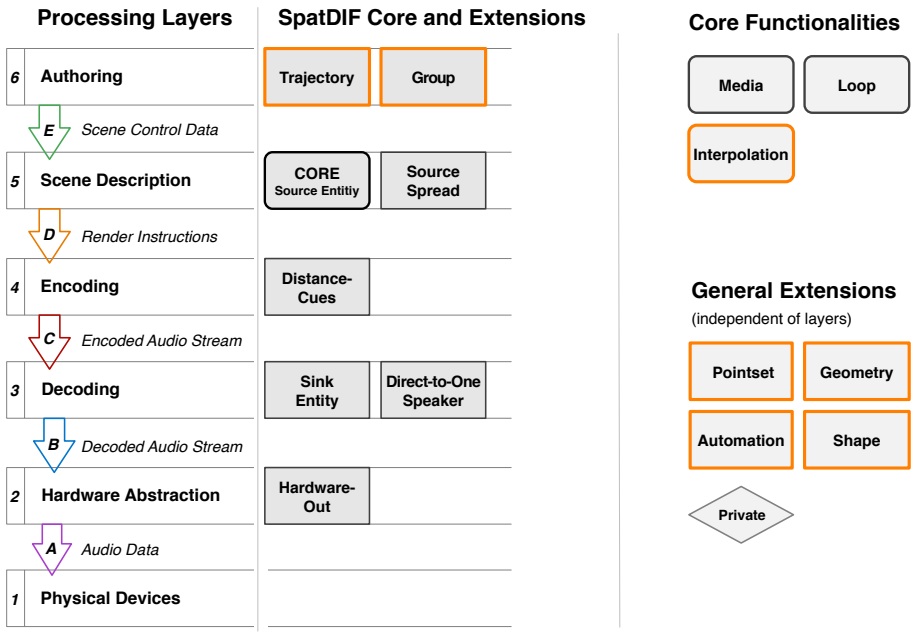


Figure 1. Left column: the processing layers and data streams in a spatialization workflow. Middle and right column: the categorization of SpatDIF extensions. New elements in SpatDIF version 0.4 are highlighted in orange and include the trajectory and group extensions on the authoring layer, a source-spread extension on the scene description layer, refined core functionalities, and four new general extensions.

cal for storing or transmitting a finished piece, where all artistic decisions have been taken and the piece just has to be rendered. It does not, however, keep any trace of how these artistic decisions and their manifestation in the scene, be it in position changes, loudness changes, etc., came to be. Also, because current discretization methods of trajectories and their perceptual artifacts is an ongoing research topic (e.g., [11]) and may improve in the future, it is desired to preserve the "master tapes" of sound trajectories along with their discretized version [12].

In other words, version 0.3 of SpatDIF is a rendering-centric syntax with a discretized representation of the audio scene.

1.2 SpatDIF Terminology

A SpatDIF representation is the combination of a space and the actions that are unfolding within it. A scene consists of a number of SpatDIF **entities**. Entities are all objects that are affecting or interacting with the sound of that scene. Entities can be of different **kinds** e.g., sources or sinks. Each entity instance is assigned a **name**, so that it may be uniquely identified within the scene. The properties of entities are described and transmitted via SpatDIF **descriptors**. A complete SpatDIF statement consists of an **address** unambiguously identifying an **entity**, its **descriptor**, and its associated **value**. The values of descriptors may change over time. All **entities** and **descriptors** are defined within the SpatDIF **namespace** which consists of core descriptors, and descriptors organised in extensions that add functions. Finally, a SpatDIF representation consists of two sections - a **meta section** and a **time section**. The meta section serves to configure and initialize the system, while the time section describes the temporal unfolding of a scene.

2. NEW IN SPATDIF VERSION 0.4

To enable the authoring of spatial music with SpatDIF the new specification version 0.4 [13] extends the scope of SpatDIF to the sixth processing layer, the **authoring layer**. Layer six describes processes that drive changes, whereas layer five contains a discrete representation of the resulting state of a scene at specific times. On this sixth layer, compositional processes that build a sound scene take place. A vast number of operations or instructions can be imagined that manipulate sound entities in the scene. Of these many possible dimensions, the descriptors at this layer currently address motion.

In combination with the already defined appearance of source entities, their media assignment and an additional source-spread factor, a spatial sound composition using popular spatialization algorithms such as VBAP [14] or DBAP [15] can be fully represented.

2.1 The Trajectory Extension

The new authoring layer in SpatDIF is defined by the trajectory extension. Applying the trajectory extension to an entity in the time-section generates a specific spatial motion in time called trajectory.

A simple example for such a description would be the following instruction in natural language: "Move source N from point A to point B by following a straight line over 5 seconds with constant speed." In technical terms, a trajectory is a result of a combination of the following functionalities:

- A shape created by a **pointset** (see Section 2.2) or a predefined **shape** template (see Section 2.5).
- Affine **geometry** transformations to manipulate this shape (see Section 2.6).

- A spatial **interpolation** method describing how to get from point to point (see Section 2.3).
- An **automation** profile, describing how a trajectory proceeds in time (see Section 2.4).

These functionalities are addressed in the interpolation core functionality and in the layer-independent extensions pointset, geometry, automation, and shape.

Since the pointset extension can serve to predefine templates and is dependent on the interpolation functionality to define a shape it can be placed either in the meta or the time-section. The automation is applied to an entity in the time-section because it affects the temporal unfolding in the scene. Both the shape templates and the geometry transformations are used in the time-section to facilitate the repeated path-description of a trajectory from templates.

The following extensions may be used independently of the authoring layer for other purposes.

2.2 The Pointset Extension

A pointset describes a group of geometrical positions or points. These can be key-points on a path, shape, curve, or polygon of any kind, but also a collection of entities of the same kind, such as sink- or speaker-positions.

Points in a pointset can be of two kinds: actual points (default), i.e., *anchors*, and helper points, i.e., *handles*. The second kind is needed to describe cubic-bezier splines, but could also serve as reference-points in other functionalities (see Section 2.3).

2.3 Extended Interpolation Functionality

For the use in trajectories, the core functionality for interpolation were extended. An interpolation defines the method used for sampling between two defined values (points/positions).

The interpolation functionality now provides three methods:

The **none** method is used to stop a motion, for example when overwriting pre-existing layer five information.

The **linear** interpolation is the default method. It can be carried out in cartesian coordinates to obtain straight lines, while linear interpolation using polar coordinates leads to arc motions (this applies to entity positions only, for arc shaped trajectories, see Section 2.5).

SpatDIF version 0.4 features the newly defined **cubic-bezier** interpolation functionality [16] as its principal way to describe curved paths and automation profiles.

Contrary to other splines, a cubic-bezier curve is defined by four points; the first and last are the anchors points that bound the curve, the middle two points (P1, P2) are handles that are used to steer the tension or curvature (see Section 2.2). This makes the use of bezier curves unambiguous and has the advantage of generating a fallback polygon of anchor-points, thus producing a shape that still resembles the original intention. In the case of a multi-segment cubic-bezier curve, the last and first point of each contiguous segment are shared, thereby reducing the number of points of the pointset approximately by a quarter [17].

2.4 The Automation Extension

The automation timing function describes how the sampling cursor moves along the path over the duration of the trajectory. The control points for the functions are described in two-dimensional relative coordinates of time over value, abbreviated `tv`. They range from 0 to 1 and go from trajectory start to trajectory end in the time relative to the duration of the motion.

Typical automation movements begin with a slow acceleration and end in a deceleration to mimic the physical behavior of objects with mass; these time profiles are called *easing* curves (see the ease-in-out function in Fig. 2). The default easing function is linear with constant speed over the entire path. A number of standard easing functions are provided that mirror the timing functions of CSS transitions [18].

The addition of multi-segment polygons or cubic-bezier curves completes the selection of functions and caters to almost all imaginable curves (bottom of Fig. 2).

By using oscillatory, zig-zag or rectangular shapes in the automation timing function, motion patterns such as palindrome looping, jumps and other interesting behaviors along the trajectory may be produced.

With these three *essential* general extensions, a trajectory can now be fully articulated in the time-section. As with any other element in a typical compositional method, however, shapes and trajectories want to be re-used and modified again and again. To avoid having to repeat a shape definition every time it is applied as an entity's trajectory, the capability to pre-define and recall shapes, pointsets and other elements is crucial. For this purpose two additional extensions have been defined.

2.5 The Shape Extension

To facilitate describing the most common trajectories, SpatDIF provides a few basic shape-primitives in the shape extension. These default shapes are defined with standardized size and orientation (see Fig. 3).

The **point** primitive serves to stop or fixate a trajectory; the **line**, **triangle** and **rectangle** primitives provide standard shapes. The **circle** primitive consists of a closed multi-segment cubic-bezier spline with predefined tension points. Note how the rectangle and circle primitive share the same anchor points, this serves as a bridge between the two shapes. The **arc** primitive is a special case, since it is defined using only the starting point, angle and radius to the centre point and the arc angle. This function is mainly intended for spiraling motions.

The addition of an arbitrary pointset in the shape extension completes the elements needed to predefine paths or timing functions in the meta-section.

To reuse shapes within a scene, shape templates can be stored in the meta section and applied to a trajectory. The pointset defining the control polygon, as well as the interpolation method are thus predefined in a standard size, to be resized when applied as a trajectory to a specific entity in the scene. When assigning the shape-template to the entity, the first point of the pointset is attached to the current position of the entity. The use of the unique name in the

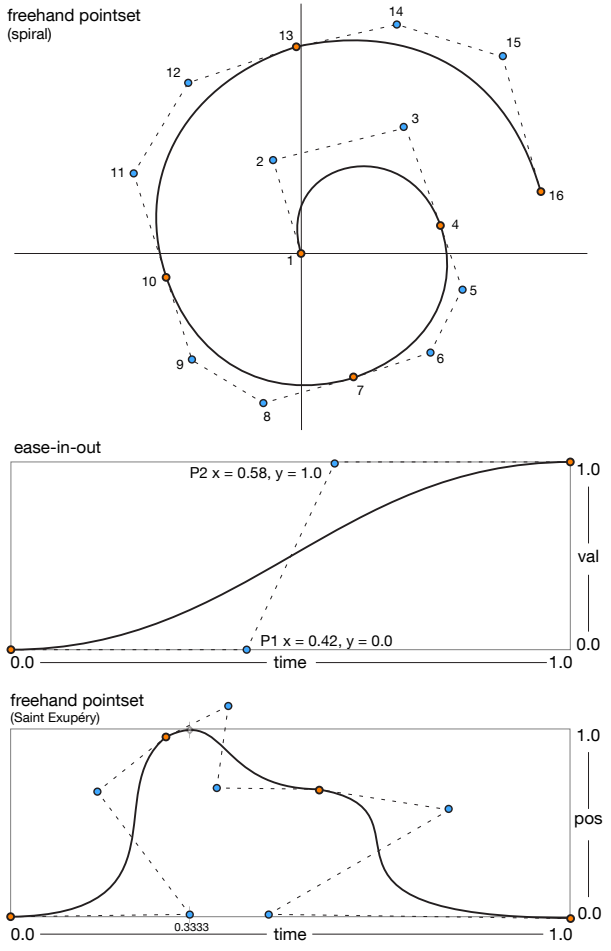


Figure 2. A trajectory consisting of a 2D spiral path (top) and two possible automation curves. The path consists of a pointset with a cubic-bezier interpolation; anchor points are marked in red, handle points in blue. The first automation curve is a predefined ease-in-out function smoothly moving from start to end of the path. The second automation curve is a free-hand multi-segment pointset with cubic-bezier interpolation. It moves from the start to the spiral's end in a third of the duration with a steep acceleration, then returns gradually to the beginning of the path in a retrograde motion.

id descriptor allows to reference a standard or pre-defined shape in the time-section [17].

When applying these shapes to an entity in a trajectory, the standard geometric properties may be transformed using the geometry extension (see 2.6).

The same pre-definition and referencing mechanism can be used for a pointset defining an automation curve.

2.6 The Geometry Extension

When applying a pre-defined shape to a trajectory, the geometric properties, such as size and orientation may need to be modified. For this purpose *affine geometrical transformations* [19] can be applied. The set of transformations includes **scale**, **translation**, **rotation**, **skew** and **mirror** and can be applied in any combination. Because the results of such transformation sequence may be order-dependent, the order of these operations is defined explicitly [17]. A

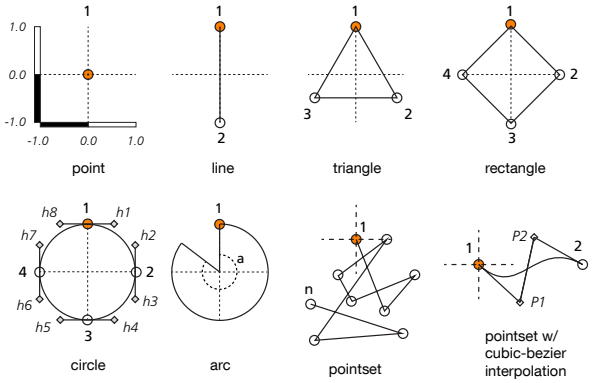


Figure 3. Predefined shape primitives, an arbitrary polygon, and a cubic-bezier curve based on the pointset.

transformed shape's first point overwrites the current position of an entity in layer five description; the entity position needs to be updated when storing the file or scene.

2.7 The Group Extension

To affect several entities at the same time the group extension is introduced on the authoring layer. This functionality can be used to apply the same behavior to a collection of entities.

This is used for compositional processes where several voices are conceptually treated as a single unit. An example might be a scene in which a vehicle is modeled that consists of the sounds of the four wheels and the engine placed at the appropriate positions in relation to each other. In order to displace the vehicle in the scene only the group's 'handle' is displaced (for example located in the driver's seat), and all other sound entities (e.g. the wheels and the engine) move by maintaining the relative position to the group (driver) (see top left of Fig. 4).

A SpatDIF group is identified and linked to by its unique name. The group represents an abstract entity and possesses the same properties and functionalities as a basic entity. For instance, it has a reference-point with a position and orientation, and this point serves as a 'handle' point for geometrical operations on the group (see top of Fig. 4). Groups can be statically defined in the meta section and/or dynamically created in the time-section. At the time of their creation, groups are initially empty. In order to populate a group, entities need to become members of a group. They attach by setting the group's unique name in their `group-membership` descriptor. An entity can only belong to a single group at a time. As long as an entity is attached to a group, the group's behavior overrides the member's behavior with respect to all descriptors that are explicitly described by the group. As a consequence attempts to change the same properties of single group members will be ignored.

When an entity joins a group, the relationship to the group is established by calculating the entity's relative (delta) value to the group's descriptor value. If a change in relationship is desired, the entity first needs to be detached from the group by setting the `group-membership` to `none`, so that it can be addressed individually, and then reattached to the group with a changed relationship, for

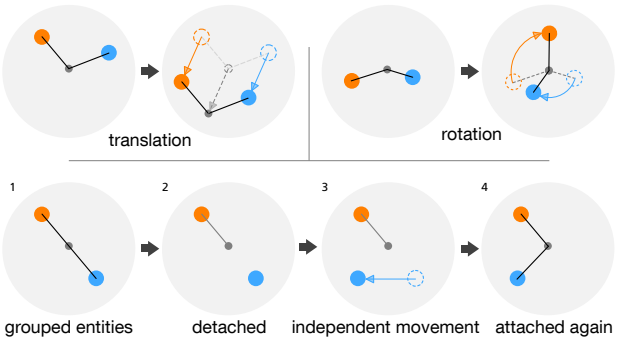


Figure 4. The group extension: A group is moved (top left) or rotated (top right); a group's member entity detaches to execute an independent movement, then attaches again (bottom).

example a shifted position or orientation, gain or spread factor (see bottom of Fig. 4). At the time of detaching from a group, the entity keeps the most recent value, including any changes that have happened as consequences of changes to the group's descriptor values. In other words, detaching from a group introduces no discontinuities in descriptor values at the fifth layer, unless a new value is set explicitly.

For descriptors affecting sound, i.e., directly active in domain of acoustics, the descriptor's unit is used to determine the manner in which to combine the group's descriptor value with the member-entities' descriptor value. When the value of a group's descriptor is changed, the change in the member-entities' descriptor-value happens according to a simple rule: If the change is expressed using a *logarithmic* unit, the group's value will be **added** to the current value of each member; if the change is expressed using a *linear* unit, the group's value will be **multiplied** with the current value of each member. For example, if the gain-level of a group is increased by 6 dB, this will increase all member-entity gain levels by 6 dB. In contrast, if the gain change is 2.0 linear units, this will double the gain levels of all member-entities.

For descriptors affecting the scene geometry, the group descriptor's value is **added** to the descriptor value of each member (see top of Fig. 4).

Core functionalities and general extensions such as trajectories, affine geometrical transforms, and automations can be used to change the group's properties instantaneously or over time. These operations can be applied to the same descriptors and in the same manner as for the entities contained within the group. As seen already, a group could be used to control the gain of several entities, rather than their position or orientation. Or the geometrical transformation could be used to rotate the entire group around the handle-point, or to shrink or expand the group by changing the scaling factors, or a combination thereof [17].

Currently, in SpatDIF v.0.4, the group extension does not support nested *hierarchies*, i.e., a group cannot contain another group [20]. Although planned for a future version, a number of open questions pertaining to the definition of this functionality still need further research and clarification.

As with any of the operations in extensions of the sixth layer, when modifying the scene, the group behaviors need to be propagated to the individual member-entities in the fifth layer description (see Section 3.1).

2.8 The Source Spread Extension

SpatDIF version 0.4 introduces one additional extension that addresses in a simple and general way the perceived spatial extent of a source.

Many rendering techniques offers methods for making the spatial localization of sources less distinct, leading to the perceptual illusion that the extent of the source spreads out. In the widely used VBAP-algorithm, for example, a width factor determines the spreading or smearing of the source across part of the sound sphere [21]. The planar DBAP algorithm has a similar blur parameter [15]. In other more advanced spatialization algorithms, source widening and diffuseness can be generated using small source motions around the position [22] or by lowering the spatial resolution in Ambisonics and other techniques to reduce directness and generate diffuseness in the sound source [23, 24].

The Source Spread Extension offers a simple and shared minimal description of the amount of spread, expressed as a percentage. The different spatial rendering processes will need to interpret this accordingly, each in relation to its spatialization principles and abilities. In general a spread of 0% should be rendered with a spatial localization that is as precise as the process is able to produce, while a spread of 100% should result in the sound being rendered in a manner that is as spread-out and non-localized as the algorithm possible can achieve.

3. DISCUSSION

Currently the descriptors on sixth layer work together with general extensions to describe source trajectories in the sound scene as well as the grouping of entities. A few fundamental rules of how to deal with this new type of representation need to be discussed.

3.1 Complementary Representations

With the introduction of the sixth layer for authoring instructions in SpatDIF version 0.4, events within the unfolding scene can be represented in two parallel ways; as layer six trajectories and as a layer five discretized, time-sampled representation. The two representations are complementary and serve slightly different purposes. The difference between them is analogues to the difference between vector-based graphics and bitmap images. Trajectories express processes, relationships and tendencies over time, and software tools for spatial composition such as Zirkonium [6] may provide intuitive graphical user interfaces for visualization and interaction with the trajectories. The ability to store trajectory information makes the resulting spatial composition more robust to future transformations such as geometric or time-related modifications, as the time-sampled representation can be recalculated to ensure adequate temporal resolution.

When making use of the trajectory extension, it may seem tempting to simply replace the layer five representation by the much more economical trajectory representation. SpatDIF requires that the layer five representation is always present in the resulting description because of the following reasons:

Direct use of the trajectory representation for rendering may impose the heavy burden of continuously interpreting the trajectories in the scene for *every* SpatDIF-compliant rendering process. The inclusion of the layer five time-sampled representation caters to relatively simple playback and spatialization processes, eliminating the need to continuously recalculate all sound properties from high-level instructions and this minimizes computation load. This also ensures that scenes authored using SpatDIF version 0.4 that make use of the trajectory extension remain backward compatible. Additionally it supports the goal of interoperability and reproduction in future software tools of unknown capability.

This does not prevent a capable software of rendering directly from a sixth layer representation. However, it imposes the presence of fifth layer information in the exported files or transmitted streams.

If a file contains sixth layer information, as indicated by the mandatory extension declaration in the meta section, a *rendering* process disregards the authoring information, whereas an *authoring* or editing process that modifies the scene's animation processes *supersedes* the simpler scene rendering information. In order to maintain the two representations in synchrony, when storing the scene to file, the modifications of the 'blueprint' of the scene in the authoring layer, i.e., of shapes that describe the evolution of scene, are always propagated down to the simpler representation, thus potentially altering and updating existing rendering instructions in layer five descriptors [17]. A further consequence of this is that if for some reason a conflicting discrepancy has emerged between the layer five and six representations of a spatial event, the layer six representation takes precedence, provided that the software reading the file is able to deal with trajectories.

3.2 The Quest for Efficiency

In his seminal analysis of sonic art Wishart provides an extended chapter dedicated to spatial motion [25, pp. 191–235]. Taking these reflections and his many concrete geometrical shape examples as a reference point, the challenge for the definition of the authoring layer descriptors lies in the bound-less variety of systems, functions and models that are capable of generating motion. Much as Wishart aims for a qualitative understanding of sound-motion in space, the SpatDIF authoring layer aims at *describing* rather than *formalizing* the resulting shape of motion generated by an algorithm or by free-hand drawing by a composer. It does *not* transport a possible formalized, mathematical representation of a source motion, such as for example the Lissajous formulas that generate the repeated figures in the canonical piece 'Turenas' by John Chowning [26]. The final resulting trajectories, however, are what the sixth layer aims at representing in their most detailed form.

This choice is done in the spirit of achieving the most with the least elements, and thus enables the methods for describing curved shapes (see Fig. 2). However, as shown in other standards such as Postscript (for example used in eps/pdf vector graphics) [27] and CSS used for rendering graphics on webpages [18], these few elements prove to be sufficiently flexible and powerful to cover all but the most exotic cases; even circular shapes can be approximated to a very high degree using multi-segment cubic-bezier curves [28].

Composers may think that SpatDIF's description of trajectories are counter-intuitive, but authoring is expected to be done with software tools that provide graphical user interfaces; hence there should be little or no need to interact directly with cubic-bezier parameter values.

In SpatDIF version 0.4 a trajectory is expressed as the combination of a spatial path and a time-based automation function. This may seem counter-intuitive compared to simply describing position in space as a function of time. In authoring tools it is however easier to access and author trajectories and movements graphically when organized in this way. In addition, this separation reflects spatial movement as it occurs in everyday life: The spatial shape, with its additional geometric transforms and spatial interpolations, describes the pathway to be followed, whereas the time-based automation function expresses how an entity or group moves along this pathway.

4. CONCLUSIONS

The recently updated version 0.4 of the SpatDIF specification [29] addresses the ability to define and store continuous trajectories on the authoring layer in a human-readable way. As a result, SpatDIF provides a new way to exchange higher level authoring data across authoring tools that helps to preserve the artistic intent in spatial music. Trajectories are described using cubic-beziers curves. With a minimum amount of functions this enables a high degree of flexibility in terms of what curves can be realized.

The new group extension enables multiple entities to be addressed collectively. In combination with trajectories this enables coordinated movements of multiple sources. Groups may also be used for mixing, where the gain level of a group can be adjusted relative to other parts of the scene while maintaining a consistent mix within the group.

SpatDIF version 0.4 compliant files that contain trajectories and groups also need to include the layer five, discretized scene description information. This ensures backwards compatibility with version 0.3 SpatDIF-compliant rendering software.

Support for SpatDIF version 0.4 authoring has already been implemented in the Zirkonium spatial audio authoring tool. Work on archiving and restoring older spatial compositions from the ZKM archive is ongoing, and the restored compositions are being saved to SpatDIF version 0.4 files [6].

Finally, in addition to improvements on the authoring layer, SpatDIF version 0.4 adds a simple description for source spread.

5. REFERENCES

[1] N. Peters, S. Ferguson, and S. McAdams, "Towards a Spatial Sound Description Interchange Format (SpatDIF)," *Canadian Acoustics*, vol. 35, no. 3, pp. 64–65, 2007.

[2] N. Peters, T. Lossius, and J. C. Schacher, "The Spatial Sound Description Interchange Format: Principles, Specification, and Examples," *Computer Music Journal*, vol. 37, no. 1, pp. 11–22, 2013.

[3] C. Miyama, G. Dipper, R. Krämer, and J. C. Schacher, "Zirkonium, SpatDIF, and mediaartbase.de: an archiving strategy for spatial music at ZKM," in *Proceedings of the Sound and Music Computing Conference*, Hamburg, Germany, 31. August – 3. September 2016.

[4] J. C. Schacher, C. Miyama, and T. Lossius, "The SpatDIF library – Concepts and Practical Applications in Audio Software," in *Proceedings of the joint International Computer Music and Sound and Music Computing Conference (ICMC|SMC|2014)*, Athens, Greece, 2014.

[5] A. Pérez-López, "Real-Time 3D Audio Spatialization Tools for Interactive Performance," Master's thesis, Universitat Pompeu Fabra, Barcelona, 2014.

[6] C. Miyama, G. Dipper, and L. Brümmer, "Zirkonium Mk III - A Toolkit for Spatial Composition," *Journal of the Japanese Society for Sonic Arts*, vol. 7, no. 3, pp. 54–59, 2015.

[7] R. Diaz and T. Koch, "Live Panorama and 3-D Audio Streaming to Mobile VR," in *AES Conference on Headphone Technology*, Aalborg, Denmark, 2016.

[8] ITU, *ITU-R BS.2076: Audio Definition Model*. Geneva, Switzerland: International Telecommunication Union, 2015.

[9] N. Peters, T. Lossius, J. C. Schacher, P. Baltazar, C. Bascou, and T. Place, "A stratified approach for sound spatialization," in *Proc. of the 6th Sound and Music Computing Conference*, Porto, PT, 2009, pp. 219–224.

[10] N. Peters, J. C. Schacher, and T. Lossius, "SpatDIF specification version 0.3, draft version," Specification of the Spatial Sound Description Interchange Format (SpatDIF) v. 0.3. 2010–2012, http://redmine.spatdif.org/projects/spatdif/files.

[11] N. Hahn, K. Choi, H. Chung, and K.-M. Sung, "Trajectory sampling for computationally efficient reproduction of moving sound sources," in *Audio Engineering Society Convention 128*, May 2010.

[12] G. Boutard and C. Guastavino, "Archiving electroacoustic and mixed music: significant knowledge involved in the creative process of works with spatialisation," *Journal of Documentation*, vol. 68, no. 6, pp. 749–771, 2012.

[13] N. Peters, J. Schacher, T. Lossius, and C. Miyama, "SpatDIF specification version 0.4, draft version," Specification of the Spatial Sound Description Interchange Format (SpatDIF) v. 0.4. 2010–2016, http://redmine.spatdif.org/projects/spatdif/files.

[14] V. Pulkki, "Virtual sound source positioning using Vector Base Amplitude Panning," *J. Audio Eng. Soc.*, vol. 45, no. 6, pp. 456–466, 1997.

[15] T. Lossius, P. Baltazar, and T. de la Hogue, "DBAP - Distance-Based Amplitude Panning," in *Proc. of 2009 International Computer Music Conference*, Montreal, Canada, 2009, pp. 489–492.

[16] M. Sarfraz, M. Asim, and A. Masood, "Capturing outlines using cubic bezier curves," in *Proceeding of the International Conference on Information and Communication Technologies*. IEEE, 2004, pp. 539–540.

[17] N. Peters, J. C. Schacher, T. Lossius, and C. Miyama, "SpatDIF Example Files," http://www.spatdif.org/examples.html.

[18] WC3 Editors, "CSS Transitions." [Online]. Available: https://drafts.csswg.org/css-transitions-1/

[19] K. Nomizu and T. Sasaki, *Affine differential geometry: geometry of affine immersions*. Cambridge University Press, 1994.

[20] J. C. Schacher, "Gesture Control of Sounds in 3D Space," in *Proceedings of the Conference on New Interfaces for Musical Expression*, New York, USA, 2007.

[21] V. Pulkki, "Uniform Spreading of Amplitude Panned Virtual Sources," in *Proc. 1999 IEEE Workshop on Applications of Signal Processing to Audio and Acoustics*, New Paltz, New York, Oct. 17-20 1999.

[22] F. Zotter, M. Frank, and M. Kronlachner, "Efficient phantom source widening and diffuseness in ambisonics," in *Proc. of the EAA Joint Symposium on Auralization and Ambisonics*, Berlin, Germany, 3-5 April 2014.

[23] T. Lossius and J. Anderson, "ATK Reaper: The Ambisonic Toolkit as JSFX plugins." in *International Computer Music Conference— Sound and Music Computing*, 2014, pp. 1338–1345.

[24] A. Sèdes, P. Guillot, and E. Paris, "The HOA Library, Review and Prospects," in *International Computer Music Conference— Sound and Music Computing*, 2014, pp. 855–860.

[25] T. Wishart and S. Emmerson, *On sonic art*. Amsterdam: Harwood Academic Publishers, 1996.

[26] J. Chowning, "Turenas: the realization of a dream," in *Proc. of the 17es Journées d'Informatique Musicale*, Saint-Etienne, France, 2011.

[27] G. Farin, *Curves and surfaces for computer-aided geometric design: a practical guide*. Elsevier, 2014.

[28] J. J. Chou, "Higher order bézier circles," *Computer-Aided Design*, vol. 27, no. 4, pp. 303–309, 1995.

[29] N. Peters, J. C. Schacher, T. Lossius, and C. Miyama, "Specification of the Spatial Sound Description Interchange Format (SpatDIF) V. 0.4," http://spatdif.org/specifications.html, 2010-2016.

THE LOOP ENSEMBLE - OPEN SOURCE INSTRUMENTS FOR TEACHING ELECTRONIC MUSIC IN THE CLASSROOM

Christof Martin Schultz
Technische Universität Berlin
c.schultz@campus.tu-berlin.de
christofmschultz@gmail.com

Marten Seedorf
Technische Universität Berlin, 3DMIN Project
marten.seedorf@campus.tu-berlin.de
drahtsalat@joyscouts.de

ABSTRACT

The electronic production, processing and dissemination of music is an essential part of the contemporary, digitalized music culture. Digital media play an important role for children and adolescents in their everyday handling of music. These types of media call for an active participation instead of mere reception and thus offer new ways of musical socialization. Despite their cultural relevance and being lively discussed in German music education, these aspects still are marginalized in the educational practice in German classrooms. In the context of the interdisciplinary research project 3DMIN, we developed the loop ensemble. It consists of three virtual instruments and is designed for the practical pedagogical dissemination of electronic music and its technical basics. The ensemble is released as an Open Educational Resource. We evaluated the instruments' usability in three ways. They were cross-checked with relevant ISO standards, three workshops were held and the participants interviewed and, finally, an accompanying analysis using the GERD model was performed, focusing gender and diversity aspects. The results show a distinct practical suitability of the ensemble, yet further empirical research is needed for a profound evaluation.

1. INTRODUCTION

This article deals with an exemplary approach to integrate digital media in general and electronic music in particular [1] into German music classrooms: The loop ensemble. In the context of the 3DMIN project [2], we developed three virtual music instruments in Pure Data. These Open Educational Resources [3] are designed as didactic material for an action-oriented music education in combination with autonomous learning, focusing electronic music culture, its aesthetics and technical principles.

2. BACKGROUND

The production, performance, storing, processing and the dissemination of music as well as listening to it are and always have been technologically determined [2]. Consequently, innovations of sound within its cultural context are often caused by technological changes [3]. In the end, the listeners are affected by these processes in various ways. The dynamic symbiosis is intensified by the progressing digitalization and in this sense digital media and their technology define the current reality of music culture [4]. Children and adolescents grow up in this cultural environment, their perception and handling of music is pervaded by digital technology. It is a distinctive characteristic of digital media, that they facilitate cultural participation in comparison to certain older forms of media. They constantly encourage users to actively shape music culture [5]. But this potential also brings challenges. For example, the users could be confronted with the complexity of a medially globalized music culture and develop a need for orientation, for a deeper understanding. The irrelevance of the own contribution within the vast mass of medial information can carry frustration. These problems illustrate the importance of media literacy in a digitalized music culture and the need of educational institutions to integrate these technocultural aspects [6].

German music education tends to be conservative. Usually, the preservation of cultural traditions is favored over the integration of cultural changes, especially if they concern medial or technological aspects of music. So, a skeptical distance from digitalization was kept for a long time [7]. Not until the beginning of the new millennium an ideological debate on basic principles led into a pragmatic handling of the topic. Impulses were sent towards educational policy, academical training and didactic practice within the classrooms, proposing feasible ways for the integration of digital media [8]. These were and are being implemented, but not extensively and often hesitantly [7]. In spite of these changes, digital music culture still is not a crucial part of German music education [9]. Yet, a fundamental willingness to update music education seems to exist, since curricula contain many formulations that stress the relevance of digital media [7, 10, 11]. The need to integrate digital music culture seems to be approved when its

[1] This abstraction is a result of the discourse within German music education, where we do not see a specific consideration of electronic music and its instruments. Instead, they are a rather marginalized part of a greater discussion, that deals with digital media in contemporary music culture and their possibilities and problems for music education as a whole.

[2] *Design, Development and Dissemination of New Musical Instruments*, an interdisciplinary between Technische Universität Berlin and Berlin University of the Arts. See http://www.3dmin.org.

[3] Open Educational Resources (OER) are „digitised materials offered freely and openly for educators, students, and self-learners to use and reuse for teaching, learning, and research" [1].

relevance in the contemporary living worlds of the pupils is considered. This basic will for change is put into perspective by the stable status quo in educational practice and this might be why the pedagogical discourse has lost its intensity since the middle of the first century [8]. Main obstacles, that were regularly described, are the teacher's lack of skills or experience dealing with digital music media, as well as the high cost and the complexity of music software [12]. As a result, younger pedagogical research in this field concentrates on the quality of learning software [8, 13, 14]. Professional music software shows severe problems when it is used in music classrooms [8] and in this regard, pedagogically adequate software is rare. There is still a need for theoretical and conceptual reflections dealing with technology, an applicable didactic and concrete proposals and examples for the use of digital media in the classroom [15].

3. THE LOOP ENSEMBLE

With the development of loop we tried to create an exemplary model of suitable music software for use in the classroom that tries to fulfill the various demands mentioned previously. As computer-based instruments loop attempts to offer the possibility to integrate digital media with a focus on electronic music culture and its technology. Loop consists of three independent but connectable electronic instruments made in Pure Data: ADD, DRUMBO and JERRY (see Fig. 1 and Fig. 2). They use different controllers, are based on different methods of sound synthesis and differentiate in their musical roles.

Figure 1: Main interface of ADD

Firstly, the issue of cost had to be taken into account. The loop instruments do this by completely relying on open source software and by being released under the General Public License (GNU GPLv3) itself, which guarantees users the freedoms to run, study, share and modify the software. The absence of licensing fees results in a zero purchase price for developers and users. Additionally, the optional hardware controllers are available for a relatively low price of approximately 50 Euros each. Open Source means flexibility and freedom in using, customizing and sharing the software. Thereby the loop ensemble meets the requirements of the current call of the German Federal Ministry of

Figure 2: Main interface of DRUMBO

Education and Research for Open Educational Resources [16].

Another urgent demand regarding music software in the classroom is the reduction of complexity [12]. Reacting to this, our basic concept of the three instruments intends that they should be able to self-describe their technical principles through interaction. To achieve this, the loop instruments are equipped with so-called illustration patches (see Fig. 3). These are subroutines, small interfaces within the interface, that use the instruments main engine but focus on a particular functionality like pitch, sequencing, reverberation, frequency modulation, amplitude modulation, ADSR envelopes, etc. With interactive minimal examples and short text blocks the patches try to explain the function of the specific modules, audio-technical basics and special phenomena. They can be opened directly from the main interface and allow the users to playfully and interactively experience individual features.

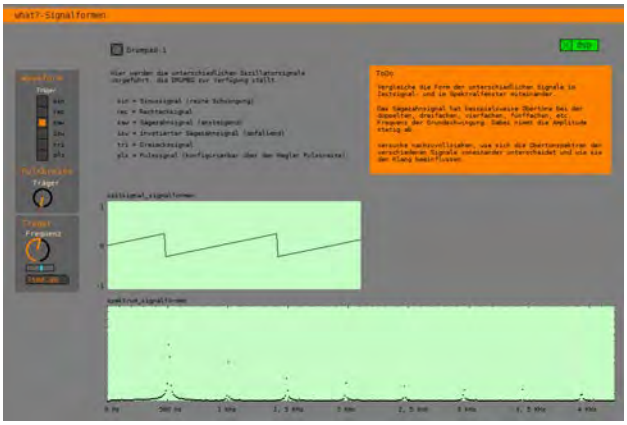


Figure 3: Illustration patch visualizing and interactively explaining waveforms within the instrument DRUMBO

The experiences obtained and conclusions drawn from the patches are meant to be used creatively when using loop as an instrument. This way the learning process is closely tied to a creative musical practice and it is this connection that makes the ensemble suitable for action-oriented lessons.

During the development we constantly had to find a balance between restricting the technical complexity in terms of the educational objectives and implementing interesting functionalities that expanded the musical possibilities.

Due to the frequently mentioned problems expressed in the pedagogical discourse we often decided to choose simplicity over functionality. Getting started with loop is particularly low-threshold. The automation established by the sequencers aids users to get started and play without a long period of training. With experience or after exploring the patches the instruments have the potential to lead to a more complex and elaborate style of play.

Connecting the three instruments with each other using the network functionality creates the ensemble, which establishes a rhythmic and harmonic synchronization that enables small groups to play together. This way loop allows intuitive cooperation and supports collective musical improvisation, making the ensemble suitable for action-oriented group lessons. This application of the ensemble is sustained by easy access and early senses of achievement, that are made possible by effective and clearly laid out controls, the optional usage of rhythmic or tonal grids, presets and sequencing: Jamming EDM with the ensemble is designed to be easy and fun, even or especially for beginners. Here ADD, DRUMBO and JERRY can loosely be assigned to musical roles: Bass, drums and pad/lead. However, these borders are quickly surmountable due to their ambivalent sound production.

The instruments are designed to support diverse traditions of electronic music. The interconnection of the three instruments enables students to play different styles of electronic dance music, depending on the adjustable aesthetic of the sounds and the settings for the rhythmic parameters. Also, loop is meant to motivate the students to experiment freely, for example by designing unusual sounds and arrangements outside of rhythmic or harmonic boundaries. Since the authors regard the breaking of barriers between musical styles as a catalyst for musical development, the loop ensemble is meant to motivate the blending of different approaches. There is no default mode. The harmonic grids and/or rhythmic synchronization can be switched on or off according to the individual user's approach and objective. Still, the possibility to produce more popular styles of electronic dance music with clear tonal and rhythmic patterns was implemented to meet the assumed preferences of the target group.

Optionally, the instruments can be controlled with low-cost hardware controllers. These are a KORG nanoKONTROL2 and an Akai LPD8. The third instrument JERRY is controlled entirely with keyboard and mouse. The graphical interfaces are adapted to the appearance of the associated controllers to help understanding their layout. All instruments can be controlled in full without the controllers using mouse and keyboard.

Due to the educational context, we tried to use appropriate language, for example with everyday analogies, that still include the technical terminologies. At the moment the ensemble only exists in German. If requested, an English version easily realizable.

The used framework Pure Data is a visual programming language that uses data flow of objects connected by patches. This is related to analog synthesizer patches and quickly enables users to get started and provides an intuitive way of programming. Pure Data is development and application environment at the same time. Building the instruments circuitry and actively using the instrument is both happening in the same window. Every change is compiled and executed in real-time, which provides direct feedback, but can also lead to deadlocks and crashes. Main advantage of Pure Data is its visual character that makes it suitable for educational use. Additionally it comes with a easy to learn and easy to use interface, plenty of libraries and a strong community. A noticeable negative effect on the usability arises from the rudimentary and limited possibilities that Pure Data offers developers for designing the user interfaces. Also the proximity of development and application environment is risky, since users can unintentionally damage primary functionalities.

To further facilitate the use of loop, the system requirements are kept to a minimum. Even ten-year-old computer hardware with any major operating system (Windows, Mac OS X, Ubuntu Linux) should be able to run the instruments smoothly. For an optimal experience and the output of lower frequencies active loudspeakers or quality headphones are highly recommended.

In summary, the didactical concept of the loop ensemble includes interactivity and stresses a self-explanatory approach. Its focus lies on self-determined and action-oriented learning. Due to its capacity to be used as an ensemble via network connection, it is also suitable for group lessons in the classroom. To support chalk-and-talk teaching as well, loop is released with an additional version, of which the interface is optimized for presentation situations. In general, the ensemble follows the formula „low threshold, high ceiling“ [17]. On the one hand loop offers an easy access to the shaping and understanding of electronic music and allows beginners or even non-musician to express themselves musically. On the other hand, it is also capable of complex musical actions and offers a deeper insight into the technical principles behind electronic sound-synthesis, for example through the exploration of the code of the instruments.

4. EVALUATION

We put the instruments through three phases of evaluation. At first we applied a rating system for music software utilized in educational contexts that uses basic ISO norms on the subject usability [8]. The results show that loop positively stands out in exploration, self-descriptiveness and suitability for learning. However, it shows deficiencies in fault tolerance and controllability in comparison with commercial products.

The second evaluation was exploratory and began while the instruments were still under development. We used them in workshops to evaluate their usability and suitability in educational contexts. For an easy access to the target group we got in contact with university support programs for girls provided by the Technische Universität Berlin and the Freie Universität Berlin. The chosen target group for the instruments, students in the upper secondary, was approached in three independent workshops (N = 10). In

those two-hour sessions the students could freely experiment with the instruments (see Fig. 4). We cautiously assisted their exploration of the ensemble by answering questions and providing suggestions. Every workshop ended

Figure 4: Full setup of loop's instrument DRUMBO with the Akai LPD8 Controller

with a group interview following a guideline. The guideline covered the subjects: Innovation, fun, usability and integration potential. The results were quite promising and provided us with valuable feedback to optimize the usability. The majority of the participants considered the instruments as desirable for their music classes. They especially valued the activity-oriented possibility to experiment and the visual presentation. The workshops also had an noticeable influence on the design and features of the instruments. The automated sequencers in ADD and DRUMBO seemed to help restrained students to start using and experimenting with the instruments. The early version of JERRY that was used in the workshops was lacking a sequencer. The students seemed to treat it with more reservation and caution. After the workshops we integrated this crucial feature to improve accessibility.

Conscious and unconscious decisions made by software developers often lead to a selective reproduction of some aspects of reality implemented in the software, while others are ignored or neglected [18]. Therefore software always has the capacity to affect sociocultural, ethical and political values and to influence the thinking and acting of users. To sufficiently deal with this, we considered a third accompanying evaluation. We used the *Gender Extended Research and Development* (GERD) analysis model, which tries to encourage developers to reflect their design choices at all critical sections of the research process and the development [19]. With its list of questions we became aware of excluded user groups, reflected about the main beneficiaries and realized how our personal background affected the development.

5. CONCLUSIONS

Ultimately, the three evaluations helped us to adopt varied perspectives and thereby improved the development of the instruments. With the valuation model of Ahlers that uses ISO usability norms we could identify innovative strengths and expectable shortcomings that came with the rudimentary open source software Pure Data. To eliminate these disadvantages we would have had to refrain from Pure Data. This would most likely have resulted in instruments that are limited in their openness, flexibility and accessibility.

The explorative workshop evaluation still showed us that the instrument appear to be usable in a classroom context. A focused evaluation of the loop ensemble in school-based practice which captures its actual suitability remains still pending. At the moment we are developing an implementation strategy at schools. We plan to organize workshops or even long-term instrumental lessons in electronic music. The didactic concepts will reflect the flexibility of the ensemble. According to our experience the best educational results are achieved with a balanced mixture of presentations (using the dedicated presentation version of the ensemble) and action-oriented sections, where the pupils are able to act and learn autonomously using the instruments. Also, we hope to be able to train teachers in the use of the ensemble and its educational content, enabling them to integrate electronic music into their lessons. Finally, we call on teachers and students to freely use, distribute and modify the loop ensemble. Loop and its manual can be downloaded free of charge at the PD Community Portal. [4]

6. REFERENCES

[1] OECD, "Giving Knowledge for Free." 2007.

[2] A. Smudits, "Musik in der digitalen Mediamorphose," in *Musik/Medien/Kunst - Wissenschaftliche und künstlerische Perspektiven*, B. Flath, Ed. Bielefeld: transcript, 2013, pp. 75–96.

[3] B. Enders, "Vom Idiophon zum Touchpad. Die musiktechnologische Entwicklung zum virtuellen Musikinstrument," in *Musik/Medien/Kunst - Wissenschaftliche und künstlerische Perspektiven*, B. Flath, Ed. Bielefeld: transcript, 2013, pp. 55–74.

[4] P. Tschmuck, "Elektronische Musik - von der Avantgarde-Nische zum paradigmatischen Musikstil," in *Musik/Medien/Kunst - Wissenschaftliche und künstlerische Perspektiven*, B. Flath, Ed. Bielefeld: transcript, 2013, pp. 97–109.

[5] M. Gall, G. Sammer, and A. de Vugt (Eds.), "Introduction to New Media in the Classroom," in *European Perspectives on Music Education: New Media in the Classroom*, ser. EAS publications. Innsbruck: Helbling, 2012, pp. 11–30.

[4] `https://puredata.info/Members/loop2016` accessed: July 11, 2016.

[6] T. Münch, "Medien im Musikunterricht," in *Musik-Didaktik: Praxishandbuch für die Sekundarstufe I und II*, W. Jank, Ed. Berlin: Cornelsen, 2013, pp. 220–228.

[7] N. Schläbitz, "Musik-Medien und die Musikpädagogik. Eine Romanze mit manchmal tragischen Momenten und ungewissem Ausgang," in preperation, in: Rolf Großmann, Elena Ungeheuer (Ed.): Musik und Medien. Laaber-Reihe: Kompendium der Musikwissenschaft, Basiswissen Musik.

[8] M. Ahlers, *Schnittstellen-Probleme im Musikunterricht: Fachhistorische und empirische Studien zum Einsatz und zur Ergonomie von Sequenzer-Programmen*, ser. Forum Musikpädagogik: Augsburger Schriften. Augsburg: Wißner, 2009, vol. 89.

[9] ——, "Information Communication Technology as Creativity Support Tools?: On German Music Educations History in ICT: Selected Research and Recent Developments," in *European Perspectives on Music Education: New Media in the Classroom*, M. Gall, G. Sammer, and A. de Vugt, Eds. Innsbruck: Helbling, 2012, pp. 125–134.

[10] Berliner Landesinstitut für Schule und Medien LISUM, "Rahmenlehrplan für die Sekundarstufe I - Musik." 2006, accessed: July 11, 2016. [Online]. Available: http://www.berlin.de/imperia/md/content/sen-bildung/schulorganisation/lehrplaene/sek1_musik.pdf

[11] ——, "Rahmenlehrplan für die gymnasiale Oberstufe - Musik." 2006, accessed: July 11, 2016. [Online]. Available: http://www.berlin.de/imperia/md/content/sen-bildung/unterricht/lehrplaene/sek2_musik.pdf

[12] G. Sammer, M. Gall, and N. Breeze, "Using music software at school: The European framework," *NET MUSIC Project 1.*, pp. 155–177, 2009.

[13] M. Stubenvoll, *Musiklernen am Computer: Zur Qualität von Musik-Lernsoftware und ihrer empirischen Überprüfung*, ser. Musikpädagogik in der Blauen Eule. Essen: Die Blaue Eule, 2008, vol. 81.

[14] B. Weidler, *Computersoftware im Musikunterricht: Am Beispiel von „Band-in-a-Box"*. Hamburg: Diplomica, 2014.

[15] C. Albers, J. Magenheim, and D. Meister, "Der Einsatz digitaler Medien als Herausforderung von Schule - eine Annäherung," in *Schule in der digitalen Welt. Medienpädagogische Ansätze und Schulforschungsperspektiven*, D. Meister, Ed. Wiesbaden: VS Verlag, 2011, pp. 7–18.

[16] Bundesministerium für Bildung und Forschung, "Richtlinie zur Förderung von Offenen Bildungsmaterialien (Open Educational Resources - OERinfo). Bundesanzeiger vom 15.01.2016," Berlin, 2016, accessed: July 11, 2016. [Online]. Available: https://www.bmbf.de/foerderungen/bekanntmachung.php?B=1132

[17] M. Resnick, B. Myers, K. Nakakoji, B. Shneiderman, R. Pausch, S. Ted, and M. Eisenberg, "Design principles for tools to support creative thinking," in *Creativity Support Tools - A workshop sponsored by the National Science Foundation*, B. Shneiderman, G. Fischer, M. Czerwinski, B. Myers, and M. Resnick, Eds. National Science Foundation, 2005, pp. 25–35, accessed: July 11, 2016. [Online]. Available: http://www.cs.umd.edu/hcil/CST/Papers/designprinciples.pdf

[18] C. Bath, "De-gendering informatischer artefakte: Grundlagen einer kritisch-feministischen technikgestaltung," Ph.D. dissertation, Fachbereich Mathematik und Informatik, Universität Bremen, 2009.

[19] C. Draude, S. Maaß, and K. Wajda, "Gender-/Diversity-Aspekte in der Informatikforschung: Das GERD-Modell," in *Gender-UseIT. HCI, Web-Usability und UX unter Gendergesichtspunkten*, N. Marsden and U. Kempf, Eds. Berlin: de Gruyter, 2014, pp. 67–78.

A SCORE-INFORMED COMPUTATIONAL DESCRIPTION OF SVARAS USING A STATISTICAL MODEL

Sertan Şentürk, Gopala Krishna Koduri, Xavier Serra
Music Technology Group, Universitat Pompeu Fabra, Barcelona, Spain
{sertan.senturk, gopala.koduri, xavier.serra}@upf.edu

ABSTRACT

Musical notes are often modeled as a discrete sequence of points on a frequency spectrum with possibly different interval sizes such as just-intonation. Computational descriptions abstracting the pitch content in audio music recordings have used this model, with reasonable success in several information retrieval tasks. In this paper, we argue that this model restricts a deeper understanding of the pitch content. First, we discuss a statistical model of musical notes which widens the scope of the current one and opens up possibilities to create new ways to describe the pitch content. Then we present a computational approach that partially aligns the audio recording with its music score in a hierarchical manner first at metrical cycle-level and then at note-level, to describe the pitch content using this model. It is evaluated extrinsically in a classification test using a public dataset and the result is shown to be significantly better compared to a state-of-the-art approach. Further, similar results obtained on a more challenging dataset which we have put together, reinforces that our approach outperforms the other.

1. INTRODUCTION

A musical note can be defined as a sound with a definite pitch and a given duration. An interval is a difference between any two given pitches. Most melodic music traditions can be characterized with a set of notes it uses and the corresponding intervals. They constitute the core subject matter of research concerning the tonality and melodies of a music system. For any quantitative analyses therein, it is required to have a working definition and a consequent computational model of notes which dictate how and what we understand of the pitch content in a music recording.

In much of the research in music analysis and information retrieval, the most commonly encountered model is one that considers notes as a sequence of points separated by certain intervals on frequency spectrum. There are different representations of the pitch content from a given recording based on this notion, the choice among which is influenced to a great degree by the intended application. Examples include pitch class profiles [1], harmonic pitch class profiles [2], pitch histograms [3] and pitch kernel density estimates [4] besides others.

Albeit a useful model of notes used alongside several information retrieval tasks, we believe it is limited in its purview. To elaborate, we consider the case of Carnatic music, an art music tradition from south India. The counterpart to note in this tradition is referred to as svara, which has a very different musicological formulation. A svara is defined to be a definite pitch value with a range of variability around it owing to the characteristic movements arising from its melodic context. The seven svaras in Carnatic music are $S(a), R(i), G(a), M(a), P(a), D(ha), N(i)$, which account for 12 pitch positions (svarasthanas), S, R_1, R_2/G_1, R_3/G_2, G_3, M_1, M_2, P, D_1, D_2/N_1, D_3/N_2, N_3 [5]. It is emphasized that the identity of a svara lies in this variability [5], which makes it evident that the former model of notes has a very limited use in this case. The arguments related to variability are also relevant to Hindustani music, an art music form prevalent in northern parts of the Indian subcontinent and as well as many other melody-dominant music cultures such as Ottoman-Turkish makam music.

In this paper, we discuss a statistical model of notes that broadens the scope of the former, encapsulating the notion of the variability in svaras (Section 3). We develop a methodology that exploits score information to automatically process the pitch content of audio recordings (Section 4). The methodology first aligns the audio recording with the relevant music score. This step is designed to handle the structural differences between the music score and the audio performance. Next, the pitch values are aggregated for each note symbol from the aligned instances of the notes and these pitch values are used to compute a statistical representation for each note. The methodology is evaluated extrinsically in a classification task comparing the results with a state-of-the-art system [6] (Section 5) using two datasets (Section 2).

Our contributions in this paper can be summarized as:

1. A novel, computational note model, which is able to describe the characteristics of the notes statistically besides its definite location
2. Adaptation of a state of the art audio-score alignment method proposed for another melody dominant culture to Carnatic music
3. Simplifications and generalizations on the adapted audio-score alignment method
4. A new dataset of Carnatic music, composed of audio recordings and music scores linked to each other in the document-level

Raaga	#Comp.	#Singer	#Rec.
Anandabhairavi	3	5	7
Atana	4	5	5
Bhairavi	5	7	8
Devagandhari	5	5	5
Kalyani	4	4	5
Todi	9	15	15
Total	**30**	**24**	**45**

Table 1. A more diverse dataset compared to the Carnatic Varnam dataset. This consists of 40 recordings in 6 raagas performed by 24 unique singers encompassing 30 compositions.

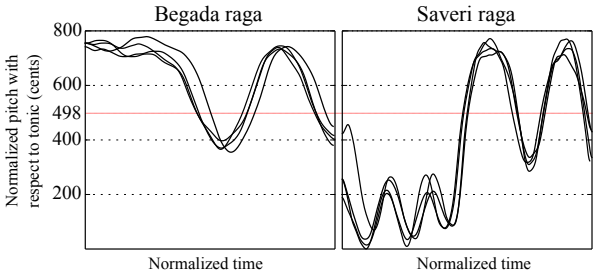


Figure 1. Example pitch contours of M_1 svara in different raagas. the X-axis is time normalized with respect to the length of each pitch contour. The tuning of M_1 svara according to the just-intomation temperament (498 cents) is indicated with a continuous red line. Notice that the majority of the pitches are sung quite distant from the theoretical tuning.

2. DATA

For evaluation, we use the Carnatic Varnam dataset [1] (see [6] for a description of varnams and the dataset). Varnams are compositions that are often sung to the score unlike several other forms which are interlaced with improvisation. Note that even though the order of the cycles in the score are retained, the performers tend to omit a few cycles or repeat a few of them twice with some minor variations. The dataset has annotations at the metrical cycle-level synchronizing the audio recording and the extracted melody with the score. There are 7 raagas, 27 recordings and 1155 cycle-level annotations. The average cycle-duration is 9.8 seconds with a standard deviation of 1.2 seconds. The music scores in the dataset are notated as a sequence of svara symbols and their relative durations. The metrical cycles are indicated in the score. There is no nominal tempo information in the score as the performance tempo is decided by the performer. With an assumption that each svara within the cycle is sung exactly according to its relative duration in the score, the svaras in the recording are annotated semi-automatically.

This dataset comes with a limitation that all the performances of a given raaga are of the same composition. Therefore the representation computed for a svara can be specific to either the raaga or the composition. In order to eliminate this ambiguity, we have put together another dataset, which is more diverse in terms of the number of compositions per raaga. [2] The details of the dataset are shared in Table 1. The Carnatic Varnam dataset is drawn from the performances of a compositional form known as varnam. Our dataset contains performances of another compositional form known as kriti. The latter are more common in concert performances, where the performers take liberty to do an impromptu improvisation. As a result, kritis are almost always not sung to the score and hence pose more challenges compared to varnams for a score-informed approach such as ours. Note that we follow the same format of the scores in Carnatic Varnam dataset to notate the kriti compositions.

[1] Available at http://compmusic.upf.edu/carnatic-varnam-dataset

[2] The dataset is available at `http://compmusic.upf.edu/node/314`.

3. MODEL OF MUSICAL NOTES

Research that involved analysis of svaras in Indian art music has time and again shown that reducing svara to a frequency value results in loss of important information [4, 7, 8]. Computational svara descriptions that use more melodic context for the description of a svara such as pitch histograms, have been shown to outperform the naive descriptions such as pitch-class distributions [6, 9]. We build on these observations from the past research and consolidate that to a statistical model of notes that would facilitate extracting information that is otherwise opaque to the currently used model.

Figure 1 shows melodic contours extracted from the individual recordings of M_1 svara (498 cents in just-intonation) in different raagas. It shows that a svara is a continuum of varying pitches of different durations, and the same svara is sung differently in two given raagas. Note that a svara can vary even within a raaga in its different contexts [7,8]. Taking this into consideration, we propose a statistical model of notes that aims for a more inclusive representation of pitches constituent in a svara. In this model, we define a note as *a probabilistic phenomenon on a frequency spectrum*. This notion can be explored in two approaches that are complementary in nature: i) *temporal*, which helps to understand the evolution of a particular instance of a svara over time (This has been theoretically explored in [8]) and ii) *aggregative*, which allows for studying the whole pitch space of a given svara in its various forms, often discarding the time information.

Our method, presented in the following section, takes the latter approach. From the annotations in our dataset, we aggregate the pitch contours over the svara reported in Figure 1 for the same set of raagas. Figure 2 shows its representations, computed as described in Section 4.2. The correspondences between the two figures are quite evident. For instance, M_1 in Begada is sung as an oscillation between G_3 (386 cents) and M_1. The representation reflects this with peaks at the corresponding places. Further, the shape of the distributions reflect the nature of pitch activity therein. The goal of our approach is to obtain such representations for svaras across different raagas in our dataset

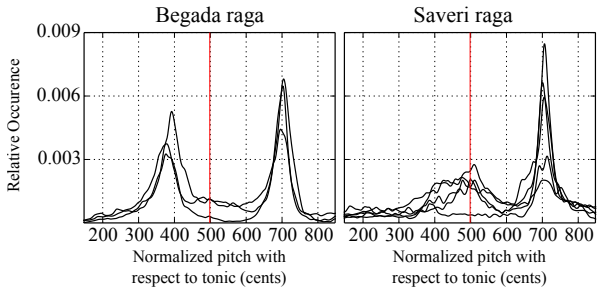


Figure 2. Histograms of M_1 svara computed from the annotated pitch contours shown in Figure 1. The tuning of M_1 svara according to the just-intomation temperament (498 cents) is indicated with continuous red lines.

automatically.

4. METHODOLOGY

Our method starts by aligning the audio and the score at the cycle- and the svara-level (Section 4.1). Then the pitch values in different instances of a given svara are obtained and an aggregate representation of a svara is computed (Section 4.2).

4.1 Audio-score alignment

Audio-score alignment can be defined as *the process of finding the segments in the audio recording that correspond to the performance of each musical element in the music score*. For this task, several approaches have been proposed using techniques such as Hidden Markov models [10, 11], conditional random fields [12] and dynamic time warping [13–15].

The structural mismatch between the music score and the audio recording is a typically encountered challenge in audio-score alignment. This is also common phenomenon in the performances of varnams and kritis, where the singers tend to repeat, omit or insert cycles in the score. To overcome this problem there exists methodologies, which allow jumps between structural elements [14, 16]. However these methodologies are not designed to skip musical events in the performance, which are not indicated in the score, such as impromptu improvisations commonly sung in kritis (Section 2). Moreover, we may not need a complete alignment between the score and audio recording in order to accumulate a sufficient number of samples for each svara.

In [17], an audio-score alignment methodology for aligning audio recordings of Ottoman-Turkish makam music with structural differences and events unrelated to the music score was introduced, and it is later extended to note-level alignment in [18]. The methodology proposed in [17], divides the score into meaningful structural elements using the editoral section annotations in the score. It extracts a predominant melody from the audio recording and computes a synthetic pitch of each structural element in the score. Then it computes a binarized similarity matrix for each structural element in the score from the predominant melody extracted from the audio recording and the synthetic pitch. The similarity matrix has blobs resembling lines positioned diagonally, indicating candidate alignment paths between the audio and the structural element in the score. Hough transform, a simple and robust line detection method [19], is used to locate these blobs and candidate time-intervals for where the structural element is performed is estimated. To eliminate erroneous estimations, [17] uses a variable-length Markov model based scheme, which is trained on structure sequences labeled in annotated recordings. Finally, Subsequence Dynamic Time Warping (SDTW) is applied to the remaining structural alignments to obtain the note-level alignment [18].

Our alignment methodology is based on the procedure described in [17, 18]. Since the original methodology is proposed for Ottoman-Turkish makam music, we optimize several parameters according to the characteristics of our data. We also modify several steps in the original methodology for the sake of generalization and simplicity. These changes will be detailed throughout this section, hereafter. The procedure in our methodology can be summarized as:

1. Extract features from the audio recording and the music score (Section 4.1.1)
2. Estimate possible partial alignments between the audio recording and the score in the cycle-level (Section 4.1.2)
3. Discard erroneous estimations (Section 4.1.3)
4. Extract svara samples from the note-level alignment within each aligned cycle. (Section 4.1.4)

4.1.1 Feature Extraction

Given an audio recording, we extract a predominant melody using the method proposed in [20], which has been shown to output reliable pitch estimations on Carnatic music recordings [6]. We denote the predominant melody extracted from the audio recording as $f = \big(f_1, \ldots, f_V\big)$, where V is the number of samples in the predominant melody. The sampling rate of the predominant melody is equal to ≈ 334.5 Hz, which is reported as an optimal for the methodology in [20]. Note that the timestamp of a pitch sample, f_i, is denoted as $\tau(f_i)$.

We then normalize the pitch values, $f_i \in f$, from Hz to cent scale with respect to the tonic frequency, t, by:

$$x_i = 1200 \log_2\Big(\frac{f_i}{t}\Big) \tag{1}$$

Note that there are 1200 cents in an octave. The tonic is extracted automatically using [21], which is reported to output near-perfect results in identifying the tonic of Carnatic music recordings. We denote the normalized predominant melody extracted from the audio recording as $x = \big(x_1, \ldots, x_V\big)$.

Parallel to audio predominant melody extraction, the svara symbols notated in the score are mapped to their cent-scale equivalents using just-intonation temperament [22]. Then, the score is divided into cycles according to the cycle boundaries annotated in the score. For each cycle (n), a synthetic pitch is computed by sampling a hypothetical continuous pitch contour corresponding to the svara sequence [17]. In this process, we consider the tempo of

the score as 70 bpm, which is reported in [9] as the average tempo in the Carnatic Varnam dataset. We denote the synthetic pitch of cycle (n) as $y^{(n)} = \big(y_1^{(n)}, \ldots, y_{W^{(n)}}^{(n)}\big)$, $n \in [1:N]$, where N is the number of cycles in the score and $W^{(n)}$ is the number of samples in the synthetic pitch. The sampling rate of the synthetic pitch is equal to the sampling frequency of the audio predominant melody. During the synthetic pitch computation, the svara onset and offset timestamps are recorded. We will use this information to obtain the svara-level alignment (Section 4.1.4) later.

4.1.2 Estimating cycle-level alignment

Instead of Hough transform used in [17], we use Iterative Subsequence Dynamic Time Warping (ISDTW) [23, Chapter 4], a common methodology used to find a queried subsequence in a given target [24, 25] to estimate the time-intervals, where a cycle is performed. Our preliminary experiments on the Carnatic Varnam Dataset showed that using ISDTW gave comparable results to Hough transform. Moreover, ISDTW simplifies the note-level alignment step compared to [18] since note onset and offsets can be directly inferred from the paths obtained from ISDTW, without introducing an additional process (e.g. SDTW in [18]) as described in Section 4.1.4.

We set the step size to $\{(2,1),(1,1),(1,2)\}$. This step size restricts the path between half and double of the tempo, which helps to avoid pathological errors. To obtain an accumulated cost matrix, $C^{(n)}$ for each cycle (n), we use the local distance measure:

$$d\left(x_i, y_j^{(n)}\right) = min\Big(\big(|x_i - y_j^{(n)}| \bmod 1200\big), \\ 1200 - \big(|x_i - y_j^{(n)}| \bmod 1200\big)\Big) \quad (2)$$

where x_i and $y_j^{(n)}$ denote the i and j^{th} samples of the audio predominant pitch x and synthetic pitch $y^{(n)}$, respectively. This distance may be interpreted as the shortest distance in cents between two pitch classes. It is not affected by octave-errors in the normalized predominant melody [17].

We use the iterative algorithm given in [23, Page 81] to estimate multiple alignments for each cycle (n). We iterate the algorithm for 10 times for each cycle. After each iteration, we obtain an estimation $e^{(k,n)}$ with an optimal alignment, $p^{(k,n)} = \big(p_1^{(k,n)} \ldots p_{L^{(k,n)}}^{(k,n)}\big)$ with $p_l^{(k,n)} = \big(r_l^{(k,n)}, q_l^{(k,n)}\big)$, $r_l^{(k,n)} \in x$, $q_l^{(k,n)} \in y^{(n)}$, $l \in \big[1 : L^{(k,n)}\big]$ (where $L^{(k,n)}$ is the length of the alignment $p^{(k,n)}$) and $k \in [1:10]$ (since there are 10 iterations for each cycle). The estimated time-interval, $t^{(k,n)}$, is the subsequence of the audio recording in the time-interval $[\tau\big(r_1^{(k,n)}\big) : \quad \tau\big(r_{L^{(k,n)}}^{(k,n)}\big)]$. For each alignment we also record the cost at each step as:

$$d^{(k,n)} = \big(d_1^{(k,n)}, \ldots, d_{L^{(k,n)}}^{(k,n)}\big) \\ = \Big(d\big(r_1^{(k,n)}, q_1^{(k,n)}\big), \ldots, d\big(r_{L^{(k,n)}}^{(k,n)}, q_{L^{(k,n)}}^{(k,n)}\big)\Big) \quad (3)$$

After each iteration, we set the values between $r_l^{(k,n)} \pm 0.1W^{(n)}$ in the accumulated cost matrix, $C^{(n)}$, to infinity for the next iteration to ensure a new path will not be searched nearby. Remember $W^{(n)}$ is the number of samples in the synthetic pitch, $y^{(n)}$.

To distinguish correct alignments from the erroneous, we compute a similarity value $s^{(k,n)} \in [0:1]$ for an iteration (k) of the cycle (n). We use the similarity measure between the cycle and the estimated alignment proposed by [17, Page 15, described as weight normalization]:

$$s^{(k,n)} = \frac{\sum_l^{L^{(k,n)}} \beta(p_l^{(k,n)}, q_l^{(k,n)})}{L^{(k,n)}} \quad (4)$$

where the binarization criteria is defined as:

$$\beta(a,b) = \begin{cases} 1, & d\big(a,b\big) \leq \alpha \\ 0, & d\big(a,b\big) > \alpha \end{cases} \quad (5)$$

In Section 5, we present the experiments to find the optimal value for the binarization threshold, α. The true positives are observed to typically emit a higher score than the erroneous ones. Performing the ISDTW for each cycle, we obtain estimations $e^{(k,n)} = \big\{n, t^{(k,n)}, p^{(k,n)}, s^{(k,n)}\big\}$, where n is the cycle extracted from the score, $t^{(k,n)}$ is the estimated time-interval in the audio recoding, $p^{(k,n)}$ is the optimal alignment of the estimation and $s^{(k,n)}$ is the similarity value of the estimation.

4.1.3 Discarding erroneous estimations

At this step we obtain a considerable number of correct estimations albeit with a comparable number of erroneous estimations. Nonetheless, we need to ensure a high precision in the cycle-level alignment to obtain a reliable svara description. In order to achieve this we can afford to trade the recall in the process since a moderate recall in the cycle-level alignment would still be able to supply a good number of samples per svara.

The method proposed for discarding erroneous estimations in [17] is not generalizable as introducing a new form with a different structure requires substantial number of training recordings in that form. For this reason, we choose to use an unsupervised estimation selection scheme, which is more generalizable and simpler.

We classify the estimations into two classes with respect to their similarity values using k-means clustering [26]. We use squared Euclidean distance as the distance measure and discard the cluster with low scores. Next, we check if there are estimations, which overlap more than 3 seconds in time. In such a case we only keep the estimation with the highest similarity value as the music has a single melody track throughout. In Section 5, we report alignment results after discarding estimations both without (i.e. only discarding overlapping estimates) and with k-means clustering.

4.1.4 Svara-level alignment

Recall that the svara onset and offset timestamps in each cycle of the synthetic pitch, $y^{(n)}$, are known. The aligned svara onset and offsets are directly obtained as the timestamps $\tau\big(r_l^{(k,n)}\big)$, which are mapped to these onsets and offsets inside the alignment $p^{(k,n)} = \big(p_1^{(k,n)} \ldots p_{L^{(k,n)}}^{(k,n)}\big)$, respectively.

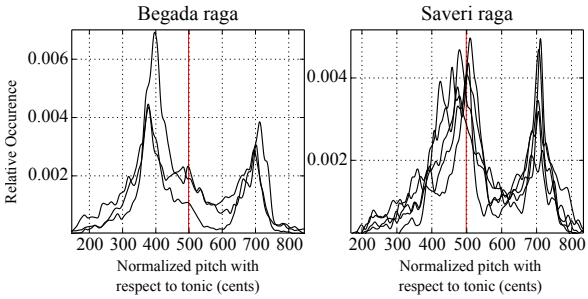


Figure 3. Description of M_1 svara (498 cents in just intonation) using our approach.

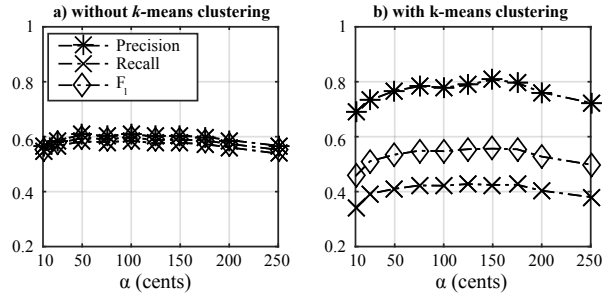


Figure 4. Results of cycle-level alignment for different binarization threshold values.

4.2 Computing svara representations

For a given recording, for each svara, σ, in the corresponding raaga, we obtain a pool of normalized pitch values, $x^\sigma = \{x_1^\sigma, x_2^\sigma, \dots\}$, aggregated over all the aligned instances from its melodic contour (Section 4.1.4). Our representation must capture the probabilities of the pitch values in a given svara. Histograms are a convenient way for representing the probability density estimates [4,6]. Therefore, we compute a normalized histogram over the pool of the pitch values. For brevity sake, we consider pitch values over the middle octave (i.e., starting from the tonic) at a bin-resolution of one cent:

$$h_m^\sigma = \frac{\sum_i \lambda_m(x_i^\sigma)}{|x^\sigma|}, \tag{6}$$

where h_m^σ is the probability estimate of the m-th bin, $|x^\sigma|$ is the number of pitch values in x^σ and λ function is defined as:

$$\lambda_m(a) = \begin{cases} 1, & c_m \leq a \leq c_{m+1} \\ 0, & \text{otherwise} \end{cases} \tag{7}$$

where a is a normalized pitch sample and (c_m, c_{m+1}) are the bounds of the m-th bin.

Figure 3 shows the representations obtained in this manner for M_1 svara (our running example from Figure 1) in different raagas. Notice that the representations obtained for M_1 are similar to the corresponding representations shown in Figure 2. This representation allows to deduce important characteristics of a svara besides its definite location (i.e., 498 cents) in the frequency spectrum. For instance, from Figure 3, one can infer that M_1 in Begada and Saveri are sung with an oscillation that ranges from G_3 (386 cents) to P (701 cents) in the former and M_1 to P in the latter.

5. EVALUATION AND RESULTS

Our method is evaluated on the two datasets described in Section 1 using the following tasks:

i. The cycle-level alignment, evaluated intrinsically using the ground truth annotations from the Carnatic Varnam dataset.

ii. The svara-level alignment and the computed representation, evaluated extrinsically using a raaga classification task on both the datasets.

The svara-level alignments cannot be verified in an intrinsic manner because marking the ground truth is prone to be erroneous as it is difficult for even musicians to agree with each other on the exact boundaries of a svara sung in a melodic continuum. [3]

To evaluate the cycle-level alignment, we check the time-distance between the estimated borders of the cycle and annotated borders as described in [17]. A cycle is marked as a true positive if the distance between both of the boundaries of the aligned cycle and the relevant annotation is less than 3 seconds. It is marked as a false positive otherwise. If there is no estimation for an annotation, it is marked as a false negative.

Figure 4 shows the recall, precision and F_1-score for different binarization thresholds used in similarity computation. Figure 4a shows that our methodology achieves a balanced recall and precision in the cycle-level alignment even without having a precise information on the performance tempo. Figure 4b shows that the process described to the discard erroneous alignments (Section 4.1.3) removes most of the false positives within an acceptable decrease in recall. It can also be observed that our cycle-level alignment is insensitive to the binarization threshold, α. When the parameter is selected between 50 cents (a quarter tone) and 200 cents (a whole tone), there is no a significant difference in the alignment results at the $p = 0.01$ level as determined by a multiple comparison test using the Tukey-Kramer statistic. Hereafter, we report results for a binarization threshold of 150 cents.

Using an α of 150 cents, we achieve a 0.42 recall, 0.81 precision and 0.56 F_1-score in cycle-level alignment after discarding the erroneous estimations. The mean and the standard deviation of the true positives are 0.62 and 0.59 seconds, respectively. Within the Carnatic Varnam dataset, we align 606 cycles and 15795 svaras in total. Out of these cycles 490 are true positives. By inspecting the false positives we observed two interesting cases: occasionally an estimated cycle is marked as false positive when one of the boundary distances is slightly more than 3 seconds. The second case is when the melody of

[3] The experiments and the results are available at `http://compmusic.upf.edu/node/314`.

Method	Carnatic Varnam dataset	Our dataset (Table. 1)
Context-based svara distributions [6]	0.62	0.64
Our approach	0.95 to 1	0.88
Using the groundtruth annotations	0.95	N/A

Table 2. Results of raaga classification task over the two datasets using different approaches.

the aligned cycle and performance is similar to each other ($s^{(k,n)} > 0.6$). In both situations considerable number of the note-level alignments would still be useful for the svara model. Within our kriti dataset, 1938 cycles and 59209 svaras are aligned in total.

We use a raaga classification task to evaluate the correctness of the svara alignments and the usefulness of the svara representation created using our statistical model. Our svara representation was shown to perform better compared to the existing representations in our previous work [6]. Therefore, in this task our primary motive is to evaluate the correctness of the svara alignments. However, as marking the svara boundaries is not a viable task, we combine it with evaluating the usefulness of the representation itself in a raaga classification task. We parametrize the representation of each svara using a set of features proposed in our aforementioned work, which include salient observations and the shape parameters of the histogram:

i. The highest probability value in the histogram of the svara
ii. The pitch value corresponding to the highest probability
iii. A probability-weighted mean of pitch values
iv. Pearson's second skewness coefficient
v. Fisher's kurtosis
vi. Variance

There are 12 svaras in Carnatic music, where each raaga has a subset of them. For the svaras absent in a given raaga, we set the features to a nonsensical value. Each recording therefore has 72 features in total.

The smallest raaga-class has three recordings in the Carnatic Varnam dataset, with few classes having more, so we subsampled the dataset six times (corresponding to the highest number of recordings for a class) with three recordings per class. We have also subsampled our dataset in a similar manner. The k-nearest neighbors classifier was earlier shown to perform the best in several raaga classification tasks with varied feature sets [6]. We use the same, with Euclidean distance metric and the number of neighbors set to one.

We compare the results of our approach with the one proposed by Koduri *et al.* [6] which was shown to outperform the previous methods of raaga classification by a slight margin. Their approach uses a moving window to estimate the local temporal context of a small section of melodic contour which is further used to estimate the svara sung at that instance. For each svara, we obtain the corresponding pitch values and use them to create a representation using the method described in Section 4.2, and parametrize it as described earlier in this section. We further compare these results with that obtained using the representation computed from the annotated svara instances in the dataset.

We performed the classification experiment over the subsamples of the two datasets using the leave-one-out cross-validation technique. For our approach, we repeated the experiment with the alignment data resulting from different binarization thresholds. The mean F_1-scores using the representations obtained from the annotations in the dataset, our approach and [6] across the subsampled datasets for the two datasets are reported in Table. 2. Our approach has performed significantly better than the earlier one in [6] on both datasets, and is on par with the method using annotated data. This is a strong indication that our description using the statistical model succeeds in capturing the variability, and therefore the identity of svaras. We also observed that different binarization threshold values have a unimportant impact on the classification accuracy.

6. CONCLUSIONS

We have presented a statistical model of musical notes that expands the scope of the current model in use by addressing the notion of variability of svaras. An approach that builds on this model and exploits scores to describe pitch content in the audio music recordings is presented and evaluated at various levels. The results clearly indicate that our approach is successful in obtaining a computational description of the svaras improving over the state-of-the-art results significantly.

The Carnatic Varnam dataset has 7 raagas, one composition per raaga sung by 3 to 5 artists. We believe this to be one of the contributing factors to a near perfect result using our approach in the raaga classification test. We have put together a more diverse dataset that encompasses more compositions per raaga. Our approach has been shown to be robust to the variability of svaras across compositions in a given raaga. However, we seek attention to the fact that our alignment method relies on the average tempo of the recordings computed from the annotations of the Carnatic Varnam dataset [6]. In order to make the system more self-reliant, we plan to add an initial tempo estimation step similar to [16] by aligning a single cycle using SDTW and resynthesizing the synthetic pitches according to the estimated tempo. We also plan to improve the alignment step by incorporating the svara models in the similarity computation within a feedback mechanism.

An interesting direction to our work is to infer possible facts about a svara from its description. For instance, answering questions such as: i) Is the svara sung steadily? ii) Where is the oscillation on a svara anchored? and so on. These can further be used as parameters that describe

the svara even more concisely. Another direction which interests us is the development of alternative computational descriptions using our statistical model of notes.

Acknowledgments

This research was partly funded by the European Research Council under the European Union's Seventh Framework Program, as part of the CompMusic project (ERC grant agreement 267583).

7. REFERENCES

[1] T. Fujishima, "Realtime chord recognition of musical sound: A system using Common Lisp Music," in *International Computer Music Conference*, 1999, pp. 464–467.

[2] E. Gómez, "Tonal description of music audio signals," Ph.D. dissertation, Universitat Pompeu Fabra, 2006.

[3] A. Gedik and B. Bozkurt, "Pitch-frequency histogram-based music information retrieval for Turkish music," *Signal Processing*, vol. 90, no. 4, pp. 1049–1063, Apr. 2010.

[4] P. Chordia and S. Şentürk, "Joint recognition of raag and tonic in North Indian music," *Journal of New Music Research*, vol. 37, no. 3, pp. 82–98, 2013.

[5] T. M. Krishna and V. Ishwar, "Karṇāṭik music: Svara, gamaka, phraseology and rāga identity," in *2nd CompMusic Workshop*, 2012, pp. 12–18.

[6] G. K. Koduri, V. Ishwar, J. Serrà, and X. Serra, "Intonation analysis of rāgas in Carnatic music," *Journal of New Music Research*, vol. 43, no. 01, pp. 72–93, Jan. 2014.

[7] M. Subramanian, "Carnatic ragam thodi – pitch analysis of notes and gamakams," *Journal of the Sangeet Natak Akademi*, vol. XLI, no. 1, pp. 3–28, 2007.

[8] A. Krishnaswamy, "On the twelve basic intervals in south Indian classical music," *Audio Engineering Society Convention*, pp. 1–14, 2003.

[9] G. K. Koduri, S. Gulati, P. Rao, and X. Serra, "Rāga recognition based on pitch distribution methods," *Journal of New Music Research*, vol. 41, no. 4, pp. 337–350, 2012.

[10] A. Cont, "A coupled duration-focused architecture for real-time music-to-score alignment," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 32, no. 6, pp. 974–987, 2010.

[11] A. Maezawa, H. G. Okuno, T. Ogata, and M. Goto, "Polyphonic audio-to-score alignment based on Bayesian latent harmonic allocation hidden Markov model," in *IEEE International Conference on Acoustics, Speech and Signal Processing*, May 2011, pp. 185–188.

[12] C. Joder, S. Essid, and S. Member, "A conditional random field framework for robust and scalable audio-to-score matching," *IEEE Transactions on Audio, Speech and Language Processing*, vol. 19, no. 8, pp. 2385–2397, 2010.

[13] S. Dixon and G. Widmer, "Match: A music alignment tool chest," in *International Society for Music Information Retrieval Conference*, 2005, pp. 492–497.

[14] C. Fremerey, M. Müller, and M. Clausen, "Handling repeats and jumps in score-performance synchronization," in *International Society for Music Information Retrieval Conference*, 2010, pp. 243–248.

[15] B. Niedermayer, "Accurate audio-to-score alignment – data acquisition in the context of computational musicology," Ph.D. dissertation, Johannes Kepler Universität, 2012.

[16] A. Holzapfel, U. Şimşekli, S. Şentürk, and A. T. Cemgil, "Section-level modeling of musical audio for linking performances to scores in Turkish makam music," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. Brisbane, Australia: IEEE, 2015, pp. 141–145.

[17] S. Şentürk, A. Holzapfel, and X. Serra, "Linking scores and audio recordings in makam music of Turkey," *Journal of New Music Research*, vol. 43, no. 1, pp. 34–52, 3 2014.

[18] S. Şentürk, S. Gulati, and X. Serra, "Towards alignment of score and audio recordings of Ottoman-Turkish makam music," in *International Workshop on Folk Music Analysis*. Istanbul, Turkey: Computer Engineering Department, Boğaziçi University, 2014, pp. 57–60.

[19] D. H. Ballard, "Generalizing the Hough transform to detect arbitrary shapes," *Pattern recognition*, vol. 13, no. 2, pp. 111–122, 1981.

[20] J. Salamon and E. Gomez, "Melody extraction from polyphonic music signals using pitch contour characteristics," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 20, no. 6, pp. 1759–1770, Aug. 2012.

[21] S. Gulati, "A tonic identification approach for Indian art music," Masters Thesis, Universitat Pompeu Fabra, 2012.

[22] J. Serrà, G. K. Koduri, M. Miron, and X. Serra, "Assessing the tuning of sung Indian classical music," in *International Society for Music Information Retrieval Conference*, 2011, pp. 263–268.

[23] M. Müller, *Information retrieval for music and motion*. Springer, 2007.

[24] M. Müller and D. Appelt, "Path-constrained partial music synchronization," in *IEEE International Conference on Acoustics, Speech and Signal Processing*. IEEE, 2008, pp. 65–68.

[25] X. Anguera and M. Ferrarons, "Memory efficient subsequence DTW for Query-by-Example spoken term detection," in *IEEE International Conference on Multimedia and Expo*. IEEE, 2013, pp. 1–6.

[26] D. J. C. MacKay, *Information theory, inference and learning algorithms*. Cambridge University Press, 2003.

COMPOSITION IDENTIFICATION IN OTTOMAN-TURKISH MAKAM MUSIC USING TRANSPOSITION-INVARIANT PARTIAL AUDIO-SCORE ALIGNMENT

Sertan Şentürk, Xavier Serra
Music Technology Group, Universitat Pompeu Fabra, Barcelona, Spain
{sertan.senturk, xavier.serra}@upf.edu

ABSTRACT

The composition information of audio recordings is highly valuable for many tasks such as automatic music description and music discovery. Given a music collection, two typical scenarios are retrieving the composition(s) performed in an audio recording and retrieving the audio recording(s), where a composition is performed. We present a composition identification methodology for these two tasks, which makes use of music scores. Our methodology first attempts to align a fragment of the music score of a composition with an audio recording. Next, it computes a similarity from the best obtained alignment. True audio-score pair emits a high similarity value. We repeat this procedure between all audio recordings and music scores, and filter the true pairs by a simple approach using logistic regression. The methodology is specialized according to the cultural-specific aspects of Ottoman-Turkish makam music (OTMM), achieving 0.96 and 0.95 mean average precision (MAP) for composition retrieval and performance retrieval tasks, respectively. We hope that our method would be useful in creating semantically linked music corpora for cultural heritage and preservation, semantic web applications and musicological studies.

1. INTRODUCTION

Version identification is an important task in music information retrieval which aims to find the versions of a music piece from a collection of audio recordings automatically [1, 2]. For popular music such as rap, pop and rock, the task aims to identify the covers of an original audio recording. For classical music traditions a more relevant task is associating compositions with the audio performances. The composition information is highly useful in many other computational tasks such as automatic content description and music discovery (e.g. searching the performances of a composition in a music collection).

For classical music cultures, music collections consisting of music scores and audio recordings along with editorial metadata are desirable in many applications involving cultural heritage archival, music preservation and musicological studies. Composition identification is a crucial step linking performances and compositions during the creation of such music corpus from unlabeled musical data [3].

Composition information can be used to generate and improve linked musical data, enhance the music content description and facilitate navigation in semantic web applications. Consider a scenario, where a musician uploads his interpretation of a composition to a platform such as SoundCloud, YouTube etc. The performed compositions can be automatically identified and labeled semantically using an ontology, e.g. [4]. Next the performance can be linked with related concepts (e.g. form, composer, score) available in other sources such as biographies of the performing artist, the music score of the composition or the musical and editorial metadata stored in open encyclopedias such as MusicBrainz and Wikipedia. Such a scheme would facilitate searching, accessing and navigating relevant music content in a more informed manner. Likewise, tasks such as enhanced listening and music recommendation may also benefit from the musical data linked via automatic composition identification.

Due to inherent characteristics of the oral tradition and the practice of Ottoman-Turkish makam music (OTMM), performances of the same piece may be substantially different from one another. This aspect brings certain computational challenges for the computational analysis and retrieval of OTMM (Section 2). In this paper, we propose a composition identification methodology, which makes use of the available music scores of the relevant compositions using partial audio-score alignment. The methodology is designed to address the culture-specific challenges brought by OTMM. To the best of our knowledge, our methodology is the first automatic composition identification proposed for OTMM. We consider two composition identification scenarios, **1)** identifying the compositions performed in an audio recording, **2)** identifying the audio recordings in which a composition is performed. Note that there might not be any relevant audio recordings for some compositions, and vice versa. Our methodology also aims to identify such cases. Our contributions can be summarized as:

1. The first composition identification methodology applied to Ottoman-Turkish makam music
2. An open and editorially complete dataset for composition identification in OTMM (Section 5.1)
3. Extending the state of the art in transposition-invariant partial audio-score alignment for OTMM by in-

troducing subsequence dynamic time warping (Section 4.2.2)
4. Simplifications and generalizations of the fragment selection and the fragment duration steps used in the score-informed tonic identification method proposed by [5] and verification of this method on a larger dataset as a side product of the composition identification experiments (Table 1)

For reproducibility purposes, relevant materials such as musical examples, data and results are open and publicly available via the Compmusic Website. [1]

The rest of the paper is structured as follows: Section 2 provides an introduction to Ottoman-Turkish makam music. Section 3 gives a definition of the composition identification tasks we are dealing with. Section 4 explains the methodology applied to both composition identification scenarios explained above. Section 5 presents the experimental setup, the test dataset and the results. Section 6 discusses the obtained results. Section 7 wraps up the paper with a brief conclusion.

2. OTTOMAN-TURKISH MAKAM MUSIC

The melodic structure of most of the traditional music repertoires of Turkey follow the concept of *makam*s [6]. Currently, Arel-Ezgi-Uzdilek (AEU) theory is the mainstream theory for OTMM [6]. AEU theory argues that there are 24 equal intervals and that a whole tone is divided into 9 equidistant intervals. These intervals can be approximated from 53-TET (tone equal tempered) intervals, each of which is termed as a *Holdrian comma* (Hc) [6].

For centuries, OMMT has been predominantly an oral tradition. Since the start of the 20th century, a notational representation extending standard Western music notation has been used in OTMM complementary to the oral practice [7]. This notation typically follows the rules of AEU theory.

Below we list some of the characteristics of OTMM, which pose challenges for composition identification:

- There is no definite tonic frequency (e.g. $A4 = 440$Hz) in the performances. The performed tonic is occasionally transposed due to instrument/vocal range or aesthetic reasons [6]. This necessitates automatic tonic identification for any fully-automatic alignment method (Section 4.2).
- The performances of OTMM occasionally include improvisations played before, after or even within a composition. It is also common to repeat, insert or omit sections of a composition.
- Until the 20^{th} century, most of these music has been strictly transmitted from a master to the students within the oral tradition. This resulted in the musical material propagating differently in different "schools." Therefore, performances of the same composition may differ from each other substantially.
- OTMM is a heterophonic music tradition. Musicians simultaneously perform the same "melodic idea;" Yet they are supposed to show their virtuosity by changing the tuning and the intonation of some intervals, adding embellishments and/or inserting, repeating and omitting notes and phrases during the performance. Melody extraction algorithms might not perform well in recordings with substantial heterophonic interactions [8].
- Most of the scores of OTMM are descriptive and they are transcribed sometimes centuries later. The scores typically notate basic, monophonic melodic lines. They do not usually indicate the heterophony, intonation deviations and other expressive elements observed in the performances.

In the experiments, we focus on *peşrev* and *saz semaisi*, which are the two most common instrumental forms of the classical repertoire. Both *peşrev* and *saz semaisi* typically consist of four non-repeating sections called *hane* and a repetitive section called *teslim* performed between these *hane*s.

3. PROBLEM DEFINITION

Given a specific music collection, two basic composition identification scenarios are:

1. **Composition retrieval:** Identification of the compositions which are performed in an audio recording.
2. **Performance retrieval:** Identification of the audio recordings in which a composition is performed.

These scenarios are ranked retrieval problems where the query is an audio recording and the retrieved documents are the compositions in the composition identification task, and vice versa. In both cases, the common step is to estimate whether a composition and an audio recording are *relevant* to one another. The relevances in the composition identification problems are binary, i.e. 1 if the composition and the audio recordings are paired and 0 otherwise.

The results in both cases can be aggregated by applying this step to multiple documents and queries. Nevertheless, there might be situations where it may be impossible or impractical to retrieve the whole collection, for example restricted access to copyrighted music material or the lack of computational resources in fast-query applications (e.g. real-time composition identification in mobile applications). Moreover, both scenarios might require different constraints to obtain better results and/or process more efficiently. For example, a good performance retrieval method should find multiple relevant audio recordings for a composition; on the other hand only the top ranked documents are important in composition retrieval when more than a single composition is rarely performed in the queried audio recordings (Section 5.1). In this paper, we deal with these two tasks separately and leave the joint retrieval task as a future direction to explore.

4. METHODOLOGY

We assume that the scores of the compositions are available and estimate the relevance by partially aligning the

[1] http://compmusic.upf.edu/node/306

score of a composition (n) with the audio recording of a performance (m). The alignment step in our methodology is based on the score-informed tonic identification procedure described in [5], which we use to obtain the best possible alignment between a score and an audio recording in a manner invariant of the transposition of the performance (Section 4.2). Next, we compute a similarity value $\in [0, 1]$ between the composition and the performance from the best alignment path. We observe a high similarity value, if the composition (n) is indeed performed in the audio recording (m) (Section 4.3). The block diagram of transposition invariant partial-audio score alignment is given in Figure 1. The alignment process is repeated between each audio recording and music score, and a similarity value is obtained for each composition and performance pair in our collection. Finally, the performance-composition pairs with low similarity values are discarded using outlier detection (Section 4.4) and the relevant pairs are obtained.

4.1 Feature Extraction

In audio-score alignment of Eurogenetic music, features which can capture the harmony such as chroma features [9, 10] are typically used. In [8], it is shown that predominant melody performs better for OTMM due the melodic nature of the music tradition. In our method we follow the melodic features proposed for audio-score alignment of OTMM in [5] and [8].

From the audio recording (m), we extract a predominant melody using a version of the methodology proposed in [11], optimized for makam music [12]. [2] The pitch precision of the predominant melody is taken as 7.5 cents ($\approx 1/3$ Hc), which is a suitable value for tracking pitch deviations in makam music [13]. The frame rate of the extracted predominant melody is downsampled from ~ 2.9 ms to ~ 46 ms, which is shown to be sufficient for audio-score alignment in OTMM [8]. We denote the predominant melody extracted from the audio recording (m) as $X^{(m)} = \left(x_1^{(m)}, \ldots, x_{I^{(m)}}^{(m)}\right)$, $m \in [1 : M]$, where M is number of audio recordings in the collection and $I^{(m)}$ is the number of samples in the audio predominant melody (Figure 1d).

From the machine readable score of the composition (n), we first pick a short fragment (either from the start of the score or from the repetition) indicated in the score. We also try different fragment durations in Section 5. Then we sample the note symbols in the note sequence of the selected fragment according to their durations in nominal tempo indicated in the score [8]. In practice, the previous note is commonly sustained in the place of a rest, so we omit the rests in the score and add their duration to the previous note [8]. The sampled symbols are mapped to the theoretical scale-degrees in cents according to the AEU theory such that the tonic symbol is assigned to 0 cents (Figure 1b). The generated synthetic pitch track has a sampling rate of ~ 46 ms, equal to the frame rate of the predominant melody. We denote the synthetic melody computed from the score of the composition (n) as $Y^{(n)} = \left(y_1^{(n)}, \ldots, y_{J^{(n)}}^{(n)}\right)$, $n \in [1 : N]$, where N is number of compositions with scores in the collection and $J^{(n)}$ is the number of samples in the synthetic melody (Figure 1b).

Notice that the unit of the pitch values in the audio predominant melody $X^{(m)}$ is Hertz, whereas the unit of the pitch values in the synthetic melody $Y^{(n)}$ is cents. For proper alignment of $Y^{(n)}$ within $X^{(m)}$ (provided that they are related with each other), $X^{(m)}$ has to be normalized with respect to the tonic frequency.

To identify the tonic, we first compute a pitch class distribution from the audio predominant melody [5]. We use kernel-density estimation to obtain a smooth pitch class distribution without spurious peaks [5]. We select the bin width of the distribution as 7.5 cents (the same as the pitch precision of the audio predominant melody) and use a Gaussian kernel with a standard deviation of 15 cents ($\approx 2/3$ Hc) so that a pitch value contributes in an interval slightly smaller than a semitone, which is reported as optimal for this task [5]. The width of the kernel is selected as 75 cents center to tail (i.e. 5 times the standard deviation) as the contribution to the samples beyond this width are redundant.

Finally we pick the peaks of the distribution as the tonic candidates [5, 13] (Figure 1e). We denote the tonic candidates for the audio recording (m) as $C^{(m)} = \{c_1^{(m)}, \ldots, C_{K^{(m)}}^{(m)}\}$, where $K^{(m)}$ is the number of peaks in the pitch class distribution (Figure 1e). Notable is that the candidates correspond to (stable) pitch classes instead of frequencies. This choice reduces the computational complexity as we will compute the "octave-wrapped" pitch distances in the alignment step (Section 4.2, Equation 2).

4.2 Transposition-Independent Partial Alignment

Assuming a candidate $c_k^{(m)}$ obtained from the pitch class distribution as the tonic frequency, we normalize each pitch sample in the audio predominant melody to cent scale by:

$$\hat{x}_i^{(m,k)} = 1200 \log_2\left(x_i^{(m)} / c_k^{(m)}\right) \quad (1)$$

Note that there are 1200 cents in an octave. We denote the predominant melody normalized with respect to the tonic candidate $c_k^{(m)}$ as $\hat{X}^{(m,k)}$. Next, we attempt to align the score fragment to the corresponding location in the audio recording by searching the synthetic melody $Y^{(m,k)}$, computed from the selected score fragment in the normalized audio predominant melody $\hat{X}^{(m,k)}$. We compare two methods for partial alignment: 1) Hough transform, and 2) Subsequence DTW.

4.2.1 Hough Transform

The Hough transform is a simple and yet effective parametric line detection method [14]. It is previously used in section-level audio-score alignment [8], tonic identification [5] and tempo estimation [15] in OTMM and found to produce comparable results to methodologies using complex models such as hierarchical hidden Markov models [15]. Nevertheless, it cannot handle extensive tempo deviations or insertions, repetitions and omission in a musical phrase since it is a linear operation.

[2] The implementation is available in `https://github.com/sertansenturk/predominantmelodymakam`

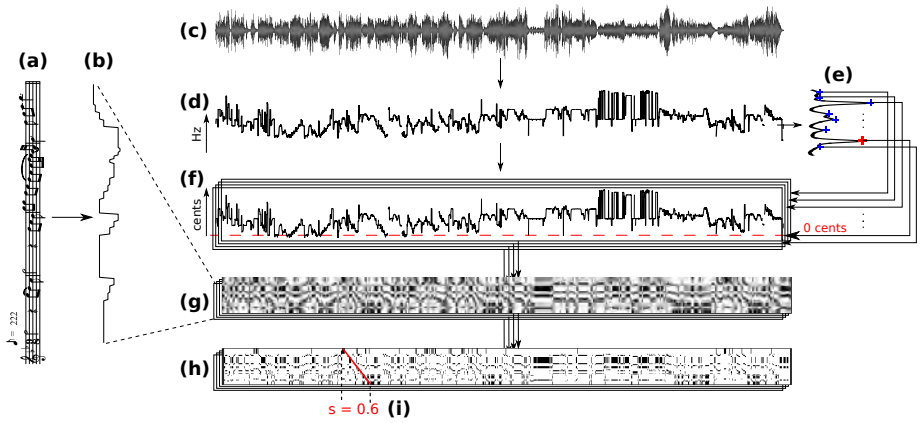


Figure 1. Block diagram of the transposition invariant partial audio-score alignment using the Hough transform **a**) A short fragment selected from the score, **b)** The synthetic pitch computed from the score fragment, **c)** The audio recording, **d)** The predominant melody extracted from the audio recording, **e)** The pitch class distribution computed from the audio predominant melody and its detected peaks, **f)** The set of predominant melodies normalized with respect to the detected peaks, **g)** The set of distance matrices between the synthetic melody and the normalized predominant melodies, **h)** The set of binary similarity matrices computed from the distance matrices. A linear alignment path obtained using the Hough transform is displayed on top of one of the binary similarity matrices along with **i)** the similarity value computed for the path. All the blocks except **g** and **h** are the same for partial alignment using SDTW.

If the Hough transform is selected for partial alignment, a distance metric is computed between the synthetic pitch track $Y^{(n)}$ and the normalized audio predominant melody $\hat{X}^{(m,k)}$. Each element $D(i,j)$ in the distance matrix D is computed as:

$$D(i,j) = min\Big(\big(|\hat{x}_i^{(m,k)} - y_j^{(n)}| \mod 1200\big), \\ 1200 - \big(|\hat{x}_i^{(m,k)} - y_j^{(n)}| \mod 1200\big)\Big) \quad (2)$$

where $\hat{x}_i^{(m,k)}$ denotes the i^{th} sample of the normalized audio predominant melody $\hat{X}^{(m,k)}$ and $y_j^{(n)}$ denotes the j^{th} sample of the synthetic pitch $Y^{(n)}$, respectively. This distance may be interpreted as the shortest distance in cents between two pitch classes. It is not affected by octave-errors in the predominant melody or the tonic.

If the selected score fragment is performed within the audio recording and the predominant melody is normalized with the correct tonic frequency, the distance matrix will show blob(s) in a diagonal trajectory formed by low distance values. The projection of the blob to the audio-axis indicates the time-interval in the audio recording where the score fragment is performed. To make the line segment more prominent, we binarize the distance matrix and obtain a binary similarity matrix B (Figure 1h). We use the binarization criteria proposed in [8] and compute each element $B(i,j)$ in the binary similarity matrix as:

$$B(i,j) = \begin{cases} 1, & D(i,j) \leq \alpha \\ 0, & D(i,j) > \alpha \end{cases} \quad (3)$$

Here two pitch values are considered to belong to the same note if the distance (in cents) is less than the given binarization threshold, α. We take $\alpha = 50$ cents, which is reported as an optimum of this value for makam music [8].

As can be seen in Figure 1g, these blobs can be approximated as line segments. To detect the line segments, we apply the Hough transform to the binary similarity matrix (Figure 1h). We restrict the searched angles between $-26.57°$ and $-63.43°$, which allows the alignment to have a tempo deviation between 0.5 and 2 times the nominal tempo indicated in the score. From the obtained transformation matrix, we select the highest peak, which indicates the most prominent line segment [14]. The linear path $p^{(m,n,k)}$, which the line segment follows, is simply the sequence of the points that has accumulated this peak in the transformation matrix. An example alignment found by the Hough transform can be seen in Figure 1h.

4.2.2 *Subsequence DTW*

Dynamic programming and more specifically dynamic time warping (DTW) are the state-of-the-art methodologies for many relevant tasks such as cover song identification [1,2] and audio score alignment [16, 17]. Unlike the Hough transform, DTW is robust to changes in tempo and musical insertions, deletions and repetitions. However, it can be prone to pathological warpings.

We use subsequence DTW (SDTW), which is a typical variant used when one of the time series is a subsequence of the other [18,19]. In this variant the paths are allowed to start/end within target. We refer the readers to [19, Chapter 4] for a thorough explanation of DTW and SDTW.

Using SDTW, we compute an element $A(i,j)$ in the accumulated cost matrix A recursively as:

$$A(i,j) = \begin{cases} 0, & i = 0 \\ +\infty, & j = 0 \\ D(i,j) + min \begin{cases} A(i-1,j-1) & \\ A(i-2,j-1), & i > 1 \\ A(i-1,j-2), & j > 1 \end{cases} & i,j \neq 0 \end{cases} \quad (4)$$

As seen above, we select the step size condition as $\{(2,1),$

$(1,1),(1,2)\}$. Analogous to the angle restriction in the Hough transform (Section 4.2.1), this step size ensures that the intra-tempo variations in any path will stay between half and double the nominal tempo indicated in the score. Moreover, we use Equation 2 as the local distance measure to calculate the accumulated cost matrix. Also, notice that the accumulated cost matrix is extended with a zeroth row and column, initialized to enable subsequence matching. Finally we back-track the path $p^{(m,n,k)}$ ending at $\arg\min_{(i)} A(i, J^{(m)})$ (remember that $J^{(m)}$ is the length of the synthetic melody), which emits the lowest accumulated cost [19, Chapter 4].

4.3 Similarity Computation

Using either the Hough transform or SDTW, we obtain a path $p^{(m,n,k)} = \left(p_1^{(m,n,k)} \dots p_{L^{(m,n,k)}}^{(m,n,k)}\right)$ between the audio recording of the performance (m) and the score of the composition (n) using the tonic candidate c_k with $p_l^{(m,n,k)} = \left(r_l^{(m,n,k)}, q_l^{(m,n,k)}\right), r_l^{(m,n,k)} \in [1, I^m], q_l^{(m,n,k)} \in [1, J^m]$ and $l \in \left[1 : L^{(m,n,k)}\right]$, where $L^{(m,n,k)}$ is the length of the path $p^{(m,n,k)}$. We compute a similarity, $s^{(m,n,k)} \in [0:1]$, between the score fragment and the audio recording for the tonic candidate $c_k^{(m)}$ by:

$$s^{(m,n,k)} = \frac{\sum_l B(r_l^{(m,n,k)}, q_l^{(m,n,k)})}{L^{(m,n,k)}} \quad (5)$$

$s^{(m,n,k)}$ gives us a measure of how closely the score fragment is followed by the corresponding time-interval in the audio recording indicated by the path. For example if the difference between the matched values of the audio predominant melody and the synthetic predominant melody are always below 50 cents, the similarity is 1.

For the partial alignment between the score of the composition (n) and the audio recording (m), we obtain a set of alignment similarities as $S^{(m,n)} = \left\{s^{(m,n,1)}, \dots, s^{(m,n,K^{(m)})}\right\}$, where $s^{(m,n,k)}$ is the alignment similarity between the composition (m) and audio recording (n) for the tonic candidate $c_k^{(m)}, k \in \left[1 : K^{(m)}\right]$.

The similarity between the composition (n) and the audio recording (m) is simply taken as the maximum alignment similarity value, i.e: $s^{(m,n)} = max(S^{(m,n)})$.

Figure 2 show the similarities computed between the performances in our audio collection (Section 5.1) and the composition, "Acemaşiran Peşrevi." [3] In this example the similarity of the relevant audio recordings are much higher compared to the non-relevant ones.

Note that finding a true pair also implies correctly identifying the tonic pitch class [5], i.e. $\zeta^{(m,n)} = \arg\max_{c_k^{(m)}} (S^{(m,n)})$, where $\zeta^{(m,n)}$ is the estimated tonic pitch class of the performance of the composition in the audio recording.

4.4 Irrelevant Document Rejection

In many common retrieval scenarios, including composition identification, the users are only interested in checking the top documents [20]. After applying partial audio-score alignment between the query and each document, we rank the documents with respect to the similarities obtained. We then reject documents with low similarities according to an automatically learned threshold.

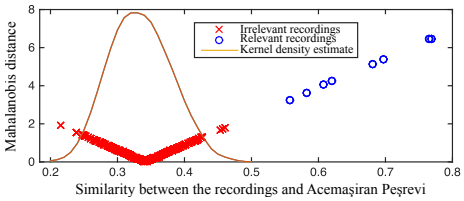


Figure 2. Similarity vs Mahalanobis distance between the composition "Acemaşiran Peşrevi" and the audio recordings in the dataset, and the kernel density-estimate computed from the similarity values between the audio recordings and the composition.

As seen in Figure 2, the relevant documents stand as "outliers" among the irrelevant documents with respect to the similarities they emit. To fetch the relevant documents per query, one can apply "outlier detection" using similarities between each document and query. Outlier detection is a common problem, which has many applications such as fraud detection and server malfunction detection [21].

Upon inspecting the similarity values emitted by irrelevant documents, we have noticed that the values roughly follow a Normal distribution (Figure 2). However, the distributions observed for each query have a different mean and variance. This is expected since the similarity computation could be affected by several factors such as the melodic complexities of the score fragment and the audio performance, as well as the quality of the extracted audio predominant melody. To deal with this variability, we compute the Mahalanobis distance of each similarity value to the distribution represented by the other similarity values (Figure 2). [4] Mahalanobis distance is a unitless and scale-invariant distance metric, which outputs the distance between a point and a distribution in standard deviations.

To reject irrelevant documents we apply a simple method where all documents below a certain threshold are rejected. To learn the decision boundary for thresholding, we apply logistic regression [20], a simple binary classification model, to the similarity values and the Mahalanobis distances on labeled data (Section 5.1). The training step will be explained in Section 5 in more detail.

After eliminating the documents according to the learned decision boundary, we add a last document called *none* to the end of the list. This document indicates that the query might not have any relevant document in the collection if all of the documents above are irrelevant.

5. EXPERIMENTS

In the experiments, we compare two alignment methods (Hough vs. SDTW). We try to align either the repetition in

[3] http://musicbrainz.org/work/01412a5d-1858-43b3-b5b0-78f383675e9b

[4] Note that the Mahalanobis distances shown in Figure 2 are less than what a "real" Normal distribution would produce. This is because of the contribution by the true pairs to the distribution.

the score as done in [5] or the start in the score as a simpler alternative and for the case when the structure information is not available in the score. We search the optimal fragment duration between 4 and 24 seconds.

As mentioned in Section 3, we evaluate the performance retrieval and the composition retrieval tasks separately. To test the document rejection step, we use 10-fold cross validation. We run the transposition-invariant partial audio score alignment between each score fragment and audio recording (Section 4.2) and then compute the similarity value for each performance-composition pair in the training set (Section 4.3). We also compute the Mahalanobis distance for each query (performance in composition retrieval task and vice versa). We apply logistic regression to the similarity values and the Mahalanobis distances computed for each annotated audio-score pair (with the binary relevances 0 or 1), and learn a decision boundary between the relevant and irrelevant documents. Then given a query (a composition in the performance retrieval task, and vice versa) from the testing set, we carry out all the steps explained in Section 4 and reject all the documents (performances in the performance retrieval task, and vice versa) "below" the decision boundary.

We use mean average precision (MAP) [20] to evaluate the methodology. MAP can be considered as a summary of how a method performs for different queries and the number of documents retrieved per query. For the document rejection step, we report the average MAP obtained from the MAPs of each testing set. We also conduct 3-way ANOVA tests on the MAPs obtained from each testing set to find if there are significant differences between the alignment methods, fragment locations and fragment durations. For all results below, the term "significant" has the following meaning: the claim is statistically significant at the $p = 0.01$ level as determined by a multiple comparison test using the Tukey-Kramer statistic.

5.1 Dataset

For our experiments, we gathered a collection of 743 audio recordings and 146 music scores of different *peşrev* and *saz semaisi* compositions. The audio recordings are selected from the CompMusic corpus [22]. These recordings are either in public-domain or commercially available. The scores are selected from the SymbTr score collection [23]. SymbTr-scores are given in a machine readable format, which stores the duration and symbol of each note. The structural divisions in the compositions (i.e. the start and end note of each section) and the nominal tempo are also indicated in the scores.

We manually labeled the compositions performed in each audio recording. In the dataset there are 360 recordings associated with 87 music scores, forming 362 audio-score pairs. This information along with other relevant metadata such as the releases, performers and composers are stored in MusicBrainz. [5] Figure 3 shows the histogram of the number of relevant compositions per audio recording and the number of relevant audio recordings per composition. The number of recordings for a particular composition in our collection may be as many as 11. On the other hand, the releases of OTMM are typically organized such that there is a single composition performed in each track. For this reason, we were only able to obtain two audio recordings in which there are two compositions performed. Note that the tonic frequency changes in the performances of each composition in these two recordings.

[5] http://musicbrainz.org/

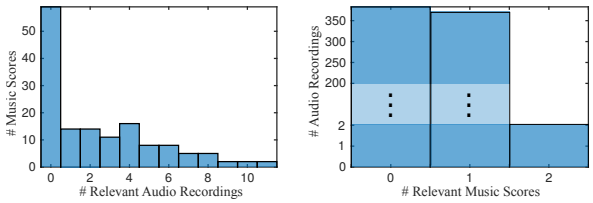


Figure 3. The number of relevant documents for the queries a) Histogram of the number of relevant audio recordings per score, b) Histogram of the number of relevant scores per audio recording

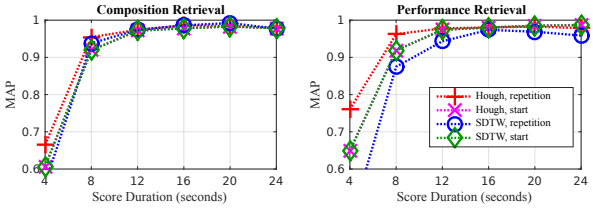


Figure 4. MAP for composition and performance retrieval task before document rejection, across different methods, fragment locations and durations. Only the queries with at least one relevant document are considered.

The average cardinalities of the compositions per audio recording and audio recordings per composition are 0.49 and 2.48, respectively. Notice that we have also included some compositions in our data collection, which do not have any relevant performances, and vice versa (Figure 3). Our methodology also aims to identify such queries without relevant documents. If we consider this case as an additional, special "document" called *none*, the average cardinality for compositions per audio recording and audio recordings per composition is 1.00 and 2.88, respectively.

5.2 Results

Before document rejection, the MAP is around 0.47 for both composition retrieval and performance retrieval tasks

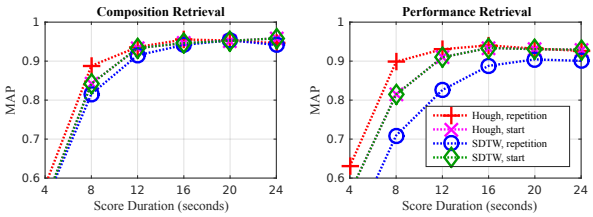


Figure 5. MAP for composition and performance retrieval task after document rejection, across different methods, fragment locations and durations. All queries are considered.

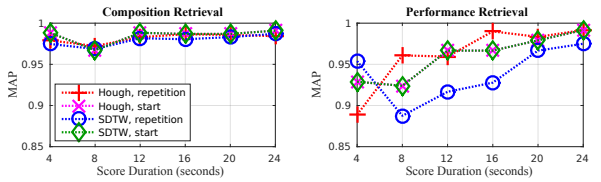


Figure 6. MAP for composition and performance retrieval task after document rejection, across different methods, fragment locations and durations. Only the queries with no relevant documents are considered.

Methods	Locations	Durations (sec.)					
		4	8	12	16	20	24
Hough	**Start**	30	15	2	3	2	2
	Repetition	14	5	0	0	0	0
SDTW	**Start**	32	6	3	3	3	3
	Repetition	24	3	1	2	3	3

Table 1. Number of errors in tonic identification

using either of the alignment methods, fragment locations and fragment durations longer than 8 seconds. The MAP is low before document rejection since the queries without relevant documents will practically have 0 average precision. Figure 4 shows the composition retrieval and performance retrieval results before document rejection only for the queries with relevant documents. The retrieval results before document rejection show that most of the audio-score pairs may be found by partial audio-score alignment by using a score fragment of at least 12 seconds. Although Hough transform performs slightly better than SDTW, these increases are not significant for fragment durations longer than 8 seconds.

Figure 5 shows the average MAPs from all queries obtained using different fragment durations, fragment locations and partial alignment methods in a 10-fold cross validation scheme. The best average MAP is 0.96 for composition retrieval using either the Hough transform or SDTW and aligning 24 seconds from the start. For performance retrieval the best average MAP of 0.95 is achieved using the Hough transform and aligning 16 seconds from the start. When we inspect average MAPs obtained from the queries without any relevant documents (Figure 6), we observe that the document rejection step always achieves an average MAP higher than 0.95 for all the parameter combinations in the composition retrieval task and an average MAP closer to or higher than 0.9 for all the parameter combinations in the performance retrieval task, respectively.

When we inspect the alignment results, we find that the score fragments were aligned properly for most of the cases. Moreover the tonic is identified almost perfectly for all the audio recordings by aligning the relevant scores (Table 1), and we achieved 100% accuracy out of the 362 audio-score pairs by aligning at least 12 seconds from the repetition using the Hough transform.

6. DISCUSSION

The results show that even aligning an 8 second fragment is highly effective, nevertheless, the optimal value of fragment duration for composition identification is around 16 seconds. Using a fragment duration longer than 16 seconds is not necessary since it increases the computation time without any significant benefit on identification performance. The results further show that aligning the start is sufficient, and there is no need to exploit the structure information to select a fragment from the repetition as in [5].

If a fragment of 16 seconds from the start of the score is selected, the Hough transform and SDTW produces the same results in both composition retrieval and performance retrieval tasks. One surprising case is the lower MAP's obtained in the performance retrieval task using SDTW to align the repetition. Although the drop is not significant for fragment durations longer than 12 seconds, we observed that SDTW tends to align irrelevant subsequences in the performances with the score fragments, which have similar note-symbol sequences but different durations.

Both the Hough transform and SDTW have a complexity of $O(I^{(m)}J^{(n)})$, where $I^{(m)}$ is the length of the predominant melody extracted from the audio recording (m) and $J^{(n)}$ is the length of the synthetic melody generated from the score of the composition (n). Nonetheless, the Hough transform is applied to a sparse, binary similarity matrix, hence it can operate faster than SDTW. Moreover, the Hough transform is a simpler algorithm. These properties make the Hough transform an alternative to more complex alignment algorithms, when precision in intra-alignment (e.g. note-level) is not necessary. Given these observations, we select alignment of the first 16 seconds of the score using the Hough transform as the optimal setting.

For the score fragments longer than 8 seconds, the tonic identification errors always occur in two historical recordings, where the recording speed (hence the pitch) is not stable and another recording where the musicians sometimes play the repetition by transposing the melodic intervals by a fifth. Even though the tonic identification has failed in these cases, the fragments are correctly aligned to the score. For such recordings, the stability of the tonic frequency can be assessed and the tonic frequency can be refined locally by referring to aligned tonic notes in the alignment path computed using SDTW.

From Figure 5, we can observe that by using a simple outlier detection step based on logistic regression, we were able to reject most of the irrelevant documents in both composition retrieval and performance retrieval scenarios. By comparing Figure 4 with Figure 5, we can also conclude that this step does not remove many relevant documents, providing reliable performance and composition matches. The usefulness of this step is more evident when the results for the queries with no relevant documents are checked (Figure 6). For such queries, since all the documents typically have a low, comparable similarity, our methodology is able to reject almost all the irrelevant documents. From Figure 6, we can also observe that the document rejection step is robust to changes in the fragment duration, the fragment location and the alignment method.

7. CONCLUSION

In this paper, we presented a methodology to identify the relevant compositions and performances in a collection consisting of audio recordings and music scores, using transposition invariant partial audio-score alignment. To the best of our knowledge, our methodology is the first automatic composition identification proposed for OTMM. The methodology is highly successful, achieving 0.95 MAP in retrieving the compositions performed in a recording and 0.96 MAP in retrieving the audio recordings where a composition is performed. What is more, our methodology is not only reliable in identifying relevant compositions and audio recordings but also identifying the cases when there are no relevant documents for a given query. Our algorithm additionally identifies the tonic frequency of the performance of each composition in the audio recording almost perfectly, as a result of partial audio-score alignment.

Our results indicate that the Hough transform can be a cheaper and effective alternative to alignment methods with more temporal flexibility such as SDTW in finding musically relevant patterns. As the next step we would like to evaluate our method on more forms, possibly with shorter structural elements such as the vocal form, *şarkı*. We would also like to investigate network analysis methods to identify the relevant performances and compositions jointly.

Our method can easily be adapted to neighboring music cultures such as Greek, Armenian, Azerbaijani, Arabic and Persian music, which share similar melodic characteristics. We hope that our method would be a starting point for future studies in automatic composition identification, and facilitate future research and applications on linked data, automatic music description, discovery and archival.

Acknowledgments

We are thankful to Dr. Joan Serrà for his suggestions on this work. This work is partly supported by the European Research Council under the European Union's Seventh Framework Program, as part of the CompMusic project (ERC grant agreement 267583).

8. REFERENCES

[1] D. P. Ellis and G. E. Poliner, "Identifying 'cover songs' with chroma features and dynamic programming beat tracking," in *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, vol. 4, 2007, pp. 1429–1432.

[2] J. Serrà, X. Serra, and R. G. Andrzejak, "Cross recurrence quantification for cover song identification," *New Journal of Physics*, vol. 11, no. 9, 2009.

[3] V. Thomas, C. Fremerey, M. Müller, and M. Clausen, "Linking sheet music and audio - Challenges and new approaches," in *Multimodal Music Processing*, ser. Dagstuhl Follow-Ups. Dagstuhl, Germany: Schloss Dagstuhl–Leibniz-Zentrum für Informatik, 2012, vol. 3, pp. 1–22.

[4] Y. Raimond, S. A. Abdallah, M. Sandler, and F. Giasson, "The music ontology," in *Proceedings of the 8th International Conference on Music Information Retrieval (ISMIR)*, Vienna, Austria, 2007, pp. 417–422.

[5] S. Şentürk, S. Gulati, and X. Serra, "Score informed tonic identification for makam music of Turkey," in *Proceedings of 14th International Society for Music Information Retrieval Conference*. Curitiba, Brazil: Pontifícia Universidade Católica do Paraná, 2013, pp. 175–180.

[6] E. B. Ederer, "The theory and praxis of makam in classical Turkish music 1910-2010," Ph.D. dissertation, University of California, Santa Barbara, September 2011.

[7] E. Popescu-Judetz, *Meanings in Turkish Musical Culture*. Istanbul: Pan Yayıncılık, 1996.

[8] S. Şentürk, A. Holzapfel, and X. Serra, "Linking scores and audio recordings in makam music of Turkey," *Journal of New Music Research*, vol. 43, no. 1, pp. 34–52, 3 2014.

[9] E. Gómez, "Tonal description of music audio signals," Ph.D. dissertation, Universitat Pompeu Fabra, 2006.

[10] M. Müller, F. Kurth, and M. Clausen, "Audio matching via chroma-based statistical features." in *Proceedings of the 6th International Conference on Music Information Retrieval (ISMIR 2005)*, 2005, p. 6th.

[11] J. Salamon and E. Gómez, "Melody extraction from polyphonic music signals using pitch contour characteristics," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 20, no. 6, pp. 1759–1770, 2012.

[12] H. S. Atlı, B. Uyar, S. Şentürk, B. Bozkurt, and X. Serra, "Audio feature extraction for exploring Turkish makam music," in *3rd International Conference on Audio Technologies for Music and Media*, Bilkent University. Ankara, Turkey: Bilkent University, 2014.

[13] A. C. Gedik and B. Bozkurt, "Pitch-frequency histogram-based music information retrieval for Turkish music," *Signal Processing*, vol. 90, no. 4, pp. 1049–1063, 2010.

[14] R. O. Duda and P. E. Hart, "Use of the Hough transformation to detect lines and curves in pictures," *Communications of the ACM*, vol. 15, no. 1, pp. 11–15, 1972.

[15] A. Holzapfel, U. Şimşekli, S. Şentürk, and A. T. Cemgil, "Section-level modeling of musical audio for linking performances to scores in Turkish makam music," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. Brisbane, Australia: IEEE, 2015, pp. 141–145.

[16] M. Müller and D. Appelt, "Path-constrained partial music synchronization," in *IEEE International Conference on Acoustics, Speech and Signal Processing*. IEEE, 2008, pp. 65–68.

[17] B. Niedermayer, "Accurate audio-to-score alignment - data acquisition in the context of computational musicology," Ph.D. dissertation, Johannes Kepler Universität, Linz, February 2012.

[18] X. Anguera and M. Ferrarons, "Memory efficient subsequence DTW for Query-by-Example spoken term detection," in *IEEE International Conference on Multimedia and Expo*. IEEE, 2013, pp. 1–6.

[19] M. Müller, *Information retrieval for music and motion*. Springer Heidelberg, 2007, vol. 6.

[20] C. D. Manning, P. Raghavan, and H. Schütze, *Introduction to information retrieval*. Cambridge University Press, 2008, vol. 1.

[21] V. Chandola, A. Banerjee, and V. Kumar, "Anomaly detection: A survey," *ACM Computing Surveys*, vol. 41, no. 3, pp. 15:1–15:58, Jul. 2009.

[22] B. Uyar, H. S. Atlı, S. Şentürk, B. Bozkurt, and X. Serra, "A corpus for computational research of Turkish makam music," in *1st International Digital Libraries for Musicology Workshop*, London, United Kingdom, 2014, pp. 57–63.

[23] K. Karaosmanoğlu, "A Turkish makam music symbolic database for music information retrieval: SymbTr," in *Proceedings of 13th International Society for Music Information Retrieval Conference (ISMIR)*, 2012, pp. 223–228.

AUTOMATIC MUSICAL INSTRUMENT RECOGNITION IN AUDIOVISUAL RECORDINGS BY COMBINING IMAGE AND AUDIO CLASSIFICATION STRATEGIES

Olga Slizovskaia
Universitat Pompeu Fabra
olga.slizovskaia@upf.edu

Emilia Gómez
Universitat Pompeu Fabra
emilia.gomez@upf.edu

Gloria Haro
Universitat Pompeu Fabra
gloria.haro@upf.edu

ABSTRACT

The goal of this work is to incorporate the visual modality into a musical instrument recognition system. For that, we first evaluate state-of-the-art image recognition techniques in the context of music instrument recognition, using a database of about 20000 images and 12 instrument classes. We then reproduce the results of state-of-the-art methods for audio-based musical instrument recognition, considering standard datasets including more than 9000 sound excerpts and 45 instrument classes. We finally compare the accuracy and confusions in both modalities and we showcase how they can be integrated for audio-visual instrument recognition in music videos. We obtain around 0.75 F1-measure for audio and 0.77 for images and similar confusions between instruments. This study confirms that visual (shape) and acoustic (timbre) properties of music instruments are related to each other and reveals the potential of audiovisual music description systems.

1. INTRODUCTION

Human perception of music is based on integrating stimuli from various modalities, mostly from the auditory and visual domains. Nevertheless, research in music description has traditionally focused on the analysis of audio recordings, without taking account of visual information [1]. The increasing availability of music videos on the internet (ex: Youtube contains a huge amount of user-generated music performances) opens the path to incorporating visual description in several music information retrieval tasks. One of the most well-established ones is musical instrument recognition, which has been researched for decades [2].

Several of the few existing works that include the analysis of both the visual and aural components are focused on the analysis and transcription movements of performer playing on percussion instruments. Marenco et al. present a method for stroke classification in audio and video recordings of Candombe drumming [3]. They employ a feature level fusion approach on edge and color filtering for drumhead, stick and hand detection from video frames and spectral features from audio. A correlation study on visual novelty and onset intensity in video recordings of drum, guitar and bass guitar performances [4] also provides some useful insights into the image processing returns for music analysis. Perhaps, the most out of ordinary idea proposed in [5] is the use of thermal imaging to detect musical instruments, while more general ones for multimodal music content analysis (including musical instrument detection) can be found in [6].

The goal of this work is to incorporate the visual modality into a musical instrument recognition system. First, we evaluate state-of-the-art image recognition techniques in the context of music instruments. Second, we reproduce the results of a state-of-the-art method for audio-based musical instrument recognition. Third, we illustrate how both approaches are integrated for musical instrument detection.

2. AUDIO AND VISUAL METHODS

2.1 Image-based musical instrument recognition

2.1.1 Selected approach

During the last years there has been a growing interest in the use of neural networks for pattern recognition. This popularity is due to several different factors. First, methods for training very deep neural networks (even with hundreds of layers) on massive datasets using the GPU for calculations have been proposed. The second reason is the success of the convolutional neural network model overcame the ImageNet 2012 image classification contest. In the last four years deep convolutional neural networks have become a standard method for image recognition, and a variety of architectures and techniques have been developed to improve the recognition accuracy. In this work we take as a basis the VGG-16 model developed by Simonyan and Zisserman [7] that demonstrated the first and the second places in the localization and classification tasks of the ImageNet ILSVRC2014 competition.

The network receives an input RGB image of size $224 \times 224 \times 3$ that is first preprocessed by substracting, for each pixel, the mean RGB value calculated from all images in the dataset. The VGG-16 network contains 16 layers of the following types: convolutional layer (CL), pooling layer (PL), fully connected layer (FC) and rectified linear units (ReLU). Furthermore, the first two fully connected layers use dropout regularization (DL) with dropout ratio set to 50%. The convolutional layers have a kernel of size

3x3 pixels and compute a dot product between the kernel and an input layer; the rectified linear units apply an elementwise activation function which is simple $\max(0, x + N(0, 1)$; the pooling layers perform a downsampling operation; the fully connected layers compute probability score, and the dropout layers thin the network to reduce overfitting. Finally, the VGG-16 model has the following architecture:

$$[Input] \rightarrow$$
$$\{[CR] \rightarrow [CRP]\} * 2 \rightarrow$$
$$\{[CR] \rightarrow [CR] \rightarrow [CRP]\} * 3 \rightarrow$$
$$\{[FRD]\} * 2 \rightarrow [FC] \rightarrow$$
$$[Probability],$$

where $[CR]$ denotes the $[CL3] \rightarrow [ReLU]$ layer, $[CRP]$ denotes the $[CL3] \rightarrow [ReLU] \rightarrow [PL]$ layer, and $[FRD]$ denotes the $[FC] \rightarrow [ReLU] \rightarrow [DL]$ layer respectively.

For our problem we use the weights of the original ImageNet pretrained VGG-16 model to initialize our network. We treat the model as a general-purpose feature extractor and retrain the last fully connected layer of the network.

2.1.2 Image dataset

For visual instrument recognition we employ a subset of the large hand-labeled ImageNet ILSVRC collection [8]. The collection originally bears 1000 classes and is intended for evaluation of image classification methods. The chosen synsets [1] are the following: accordion, banjo, cello, drum, flute, guitar, oboe, piano, saxophone, trombone, trumpet, violin.

There is a total of 19593 images of 12 classes, including images with a single instrument or with other objects around.

2.2 Audio-based musical instrument recognition

2.2.1 Selected approach

For audio classification we use a standard bag-of-features pipeline. As a baseline we select the approach from [9]. Following the steps in [9] we split audio files with a fixed framesize of 46 ms and hopsize of 24 ms using a Blackman-Harris windowing function, extract a big amount of spectral, cepstral and tonal descriptors (such as spectral centroid, spectral spread, spectral energy, pitch confidence, pitch salience etc.) and compute commonly used statistical measures (e.g., mean, variance and standard deviation) from both the actual and the delta values as described in the previous work [10]. We utilize the Essentia [11] library for feature extraction. Then we normalize all attributes using $L2$ normalization and perform χ^2 feature selection as preprocessing techniques. The original method proposes Support Vector Machine (SVM) algorithm for the final classification. We were also interested in an evaluation of scalable boosted decision trees (XGBoost) implemented in [12] due to its high performance as the winning solutions from Kaggle [2] and KDDCup [3] challenges.

[1] Corresponding wnids: n02672831, n02787622, n02992211, n03110669, n03249569, n03372029, n03467517, n03838899, n03928116, n04141076, n04487394, n04536866

	IRMAS	RWC	ImageNet
Classes	11	45	12
Samples p/class, median	626	43	1675
Samples p/class, std	125	64	125

Table 1: Summary of datasets used for instrument recognition evaluation.

2.2.2 Audio dataset

For evaluating musical instrument recognition in audio, we use two standard music collections, as detailed below. We also considered the University of Iowa Musical Instrument Samples (UIOWA MIS) [13], composed of 2182 samples of 20 instruments and referenced in the literature as a baseline dataset, obtaining 100% precision and recall.

IRMAS. This dataset includes musical audio excerpts from more than 2000 recordings in various styles and genres with annotations of the predominant instrument present. It was used for the evaluation in [9] and originally compiled for [10]. We use the training part of the collection that contains 6705 audio files in 16 bit stereo wav format sampled at 44.1kHz. They are excerpts of 3 seconds for 11 pitched instruments such as cello, clarinet, flute, acoustic guitar, electric guitar, organ, piano, saxophone, trumpet, violin, and additionally human singing voice. The median number of samples per class is 626 with standard deviation 125.

RWC Musical Instrument Sound. In our evaluation we also use the Real World Computing (RWC) Music Database: Musical Instrument Sound [14]. It contains 3544 audio excerpts labeled in 50 pitched and percussion instruments, and human voice. We take only classes that contain more than 20 objects since the original frequency distribution of the data makes difficult to perform the standard cross-validation procedure. Eventually, 45 instrument and voice classes are selected for the evaluation including piano, electric piano, glockenspiel, marimba, accordion, harmonica, classic guitar, ukulele, acoustic guitar, mandolin, electric guitar, electric bass, violin, viola, cello, contrabass, harp, timpani, trumpet, trombone, horn, soprano sax, alto sax, tenor sax, baritone sax, English horn, bassoon, clarinet, piccolo, flute, recorder, shakuhachi, shamisen, Japanese percussion, koto, concert drums, rock drums, jazz drums, percussion, soprano voice, alto voice, tenor voice, baritone voice, bass voice, R&B vocal).

Summary statistics on the datasets with the number of samples per class can be obtained in Table 1.

2.3 Multimodal fusion techniques

There are two main strategies used to integrate informa-

[2] https://www.kaggle.com/competitions

[3] http://www.kdd.org/kdd-cup

tion from several sources into a joint multimodal system: early fusion, also known as feature level fusion, and late fusion, also known as decision level fusion. In the first case, all features from different data modalities are incorporated into a large single vector for further training. In the second case, data from different sources is used for training independently and the integration is performed on the final prediction stage. Compared with early fusion, late fusion is easier to implement, has lower computational complexity and has been shown effective in practice [15]; while early fusion looks more natural from a perceptual point of view. Furthermore, early fusion requires to use a general classifier, while late fusion let us use classification methods which are more tailored to each modality.

In our case studies we follow late fusion and consider audio and visual sources independently combining them on a single frame decision level. The code of experiments, audio-based pretrained models and features are available online [4] . The finetuned VGG-16 network is available upon request.

2.4 Evaluation strategy

We first evaluate the performance of individual audio/image classifiers using standard metrics such as precision, recall and F1-score.

For the evaluation of the image-based recognition system we use stratified 5-fold cross-validation to get the average overall accuracy. Additionally, we split each train subset into the indeed training subset and the validation subset in the proportion 3:1. For each fold we select the model with the best classification accuracy on the validation subset and then evaluate on the test subset. Finally, we use a total of 11756, 2939 and 3919 images for train, validation and test sets for each fold respectively.

In order to compare the performance of the two audio-based classifiers, SVM and XGBoost, on the same dataset we follow the approach described below:

- we divide the dataset into 10 subsets for stratified 10-fold cross-validation;
- we perform multi-dimentional grid search to find the best performing combination of hyperparameters;
- once parameters are optimized, we apply the classification method and evaluate the accuracy on each subset;
- overall accuracy is averaged across all partitions; we also use these values to measure the statistical difference between classifiers;
- to compare algorithms, we use the McNemar's test as described in [16]. For each sound excerpt in the test subset, we record how it was classified by classifiers f_A and f_B and construct the following contingency table:

[4] https://github.com/Veleslavia/SMC2016

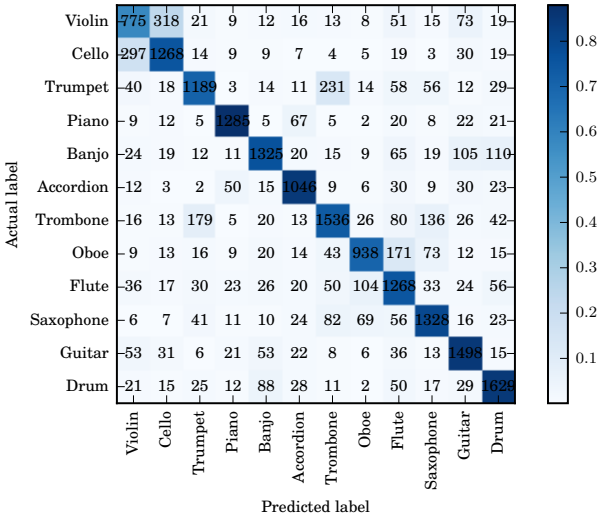


Figure 1: Confusion matrix for ImageNet musical instrument subset.

n_{00} – number of examples misclassied by both f_A and f_B	n_{01} – number of examples misclassied by f_A but not by f_B
n_{10} – number of examples misclassied by f_B but not by f_A	n_{11} – number of examples misclassied by neither f_A nor f_B

where $n = n_{00}+n_{01}+n_{10}+n_{11}$ is the total number of excerpts in the test subset.
Under the null hypothesis, the two algorithms should have the same error rate, which means that $n_{01} = n_{10}$. The McNemar's test is based on the χ^2_{MN} test statistic:

$$\chi^2_{MN} = \frac{(\mid n_{01} - n_{10} \mid) - 1^2}{n_{01} + n_{10}}$$

Next, χ^2_{MN} is compared by to the χ^2 statistics. If χ^2_{MN} exceeds $\chi^2_{1,1-\alpha}$ statistic, then we reject the null hypothesis (in our case, SVM classifier and XGBoost classifier perform equivalently on the same dataset) with $1 - \alpha$ confidence.

3. RESULTS AND DISCUSSION

3.1 Experimental results

3.1.1 Image classification

We observe in Table 2 that the overall performance is 0.77 F1 for both validation and test sets. Piano is the best classified instrument in both validation (0.88 F1) and test (0.88 F1) sets, followed by banjo, guitar, accordion and drum. Violin and flute yield the poorer performances, around 0.6 and 0.71 respectively. Figure 1 shows that the most relevant confusions correspond to instruments from the same family such as trumpet vs trombone, flute vs oboe, saxophone vs trombone or guitar vs banjo. This result is not surprising as they share similar shapes.

Instrument	Val Prec	Val Rec	Val F1	Test Prec	Test Rec	Test F1
Violin	*0.62*	*0.59*	*0.60*	*0.60*	*0.58*	*0.59*
Cello	0.76	0.79	0.77	0.73	0.75	0.74
Trumpet	0.78	0.73	0.75	0.77	0.71	0.74
Piano	**0.88**	**0.88**	**0.88**	**0.89**	**0.88**	**0.88**
Banjo	**0.82**	0.75	0.78	**0.83**	0.76	0.80
Accordion	0.78	0.85	0.82	0.81	**0.85**	**0.83**
Trombone	0.75	0.73	0.74	0.77	0.73	0.75
Oboe	0.78	*0.67*	0.72	0.79	*0.70*	0.74
Flute	*0.67*	0.75	*0.71*	*0.67*	0.75	*0.71*
Saxophone	0.78	0.77	0.78	0.78	0.79	0.79
Guitar	0.81	**0.87**	**0.83**	0.80	**0.85**	0.82
Drum	0.81	0.84	0.82	0.81	**0.85**	**0.83**
Overall	0.77	0.77	0.77	0.77	0.77	0.77

Table 2: Validation and test performances of finetuned VGG-16 CNN method on ImageNet musical instrument subset.

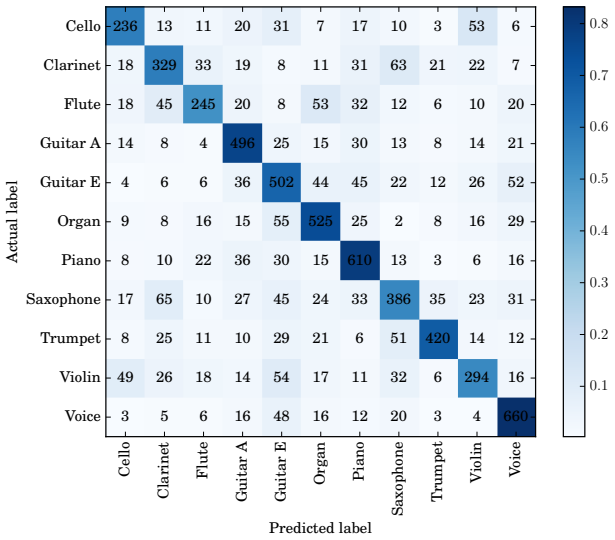


Figure 2: Confusion matrix for IRMAS dataset.

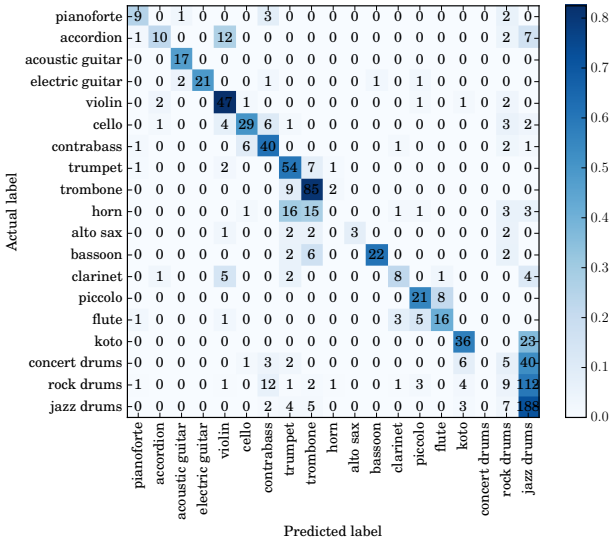


Figure 3: Confusion matrix for RWC dataset (SVM classifier).

3.1.2 Audio classification

We observe in Table 3 that XGBoost outperforms SVM approach for IRMAS dataset, with an accuracy of 0.67 (F1). With this approach and in this dataset, Voice is the best classified instrument (0.79 F1), followed by Piano (0.75 F1), which was also the best classified instrument in the image-based approach (there was no voice class in the image dataset). Violin and flute are some of the instruments with lower accuracy (0.58 F1), as it happened for the image dataset. In this approach, the saxophone has also a low accuracy, which contrasts with the image results. Figure 2 shows that the most relevant confusions correspond to instruments from the same family such as violin vs cello.

For RWC (see Table 4), XGBoost also outperforms SVM approach, with an overall accuracy of 0.83 (F1). With this approach and in this dataset, clarinet is the best classified instrument (0.79 F1). Drums is the worst classified instrument (0.58 F1). Figure 3 reveals a high confusion rate between the three classes of drums.

Although it is difficult to directly compare the results obtained from heterogeneous sources from different databases, the results are competitive with [9] and [7]. We significantly improved the classification performance with XGBoost algorithm for both audio datasets (the null hypothesis is rejected at the 0.01 significance level). We found similar confusions concerning musical instruments from the same family but there seem to be differences and similarities in the way the different instruments are distinguished through audio and visual descriptors.

3.2 Case study for combined audio and image classification

To identify musical instruments in a video we use a single-frame model from [17]. We extract image frames from video and synchronized audio excerpts of 3 seconds from corresponding audio signal. We employ the finetuned VGG-16 model to classify image frames and the IRMAS-trained XGBoost model to classify audio frames. Figure 4 illustrates an example of the results obtained for a selected set of video frames [5, 6, 7, 8].

We now provide detailed comments to each video frame. Figure 4a shows an example of the best prediction for both audio and video modalities. We observe close results in figure 4b with the high probability of the visual detection and the same confusions in the top-2 prediction as those found in the complementary confusion matrix 1. The audio has lower quality and less satisfying, although we have background voice in the first audio frame and low classification confidence in the second audio frame. We consider the next accordion example 4c as a visual-only problem since the audio classifier does not have a suitable class la-

[5] https://youtu.be/mMl_P7zVrQw?t=23
[6] https://youtu.be/eeri7gE3ZJ0?t=17
[7] https://youtu.be/jjj0Ju3mDFk?t=31
[8] https://youtu.be/J2URcUQSpv4?t=24

Instrument	SVM Prec	SVM Rec	SVM F1	XGB Prec	XGB Rec	XGB F1
Cello	0.51	*0.22*	*0.31*	*0.61*	0.58	0.60
Clarinet	*0.43*	0.48	0.45	*0.61*	0.59	0.60
Flute	**0.77**	*0.22*	0.34	0.64	*0.52*	*0.58*
Guitar ac.	0.51	0.58	0.54	0.70	0.77	0.73
Guitar el.	0.44	0.61	0.51	0.60	0.66	0.63
Organ	0.53	0.64	0.58	0.70	0.74	0.72
Piano	*0.43*	0.70	0.53	0.72	0.79	0.75
Saxophone	0.47	0.27	0.34	0.62	0.55	*0.58*
Trumpet	0.72	0.48	0.58	**0.80**	0.69	0.74
Violin	0.57	0.42	0.48	*0.61*	0.55	*0.58*
Voice	0.63	**0.75**	**0.69**	0.76	**0.83**	**0.79**
Overall	0.54	0.52	0.50	**0.68**	**0.68**	**0.67**

Table 3: Performance of the state-of-the-art SVM method compared to the XGBoost algorithm on IRMAS dataset.

bel. The image quality and confusions seem appropriate, and may be related to the fact that they have almost the same appearance of keyboard. ImageNet confusion matrix 1 also confirms this assumption. The recognition performance on the latest example 4d seems worse than expected. Nevertheless, each frame contains the correct label in the top-2 prediction of the classifiers.

Additionally, it is worthy to mention that the pattern recognition with convolutional neural networks can be challenging even for two very similar frames as confirmed in [18].

In the presented examples, we obtained a worse generalization ability for audio than for images. It can be partially explained by the high quality of the training image dataset, while real-world audio excerpts contain a lot of background noise and low-level features have been found not to be robust even to small modifications [19].

4. CONCLUSIONS

In this article, we have studied the quality of image classification and audio classification in musical instrument recognition for several datasets. Despite the difficulties associated with direct comparison of the performance obtained from heterogeneous datasets we have shown state of the art results in both modalities. Moreover, we evaluated and compared the performance of two audio classifiers and outperformed state of the art. In addition we have demonstrated the integrated single-frame method applied for real-world video recording of a musical performance.

In future work we intend to create an annotated video dataset for musical instrument detection, investigate convolutional neural networks approach for spatio-temporal feature learning in both sound and video components and explore techniques for generating audio-visual description of performance recordings.

Acknowledgments

This research was partially supported by the Spanish Ministry of Economy and Competitiveness under the María de Maeztu Units of Excellence Programme (MDM-2015-0502).

We thank Juanjo Bosch for assistance with IRMAS dataset and providing the code for feature extraction, and Marius Miron for useful comments about convolutional neural networks.

Instrument	SVM Prec	SVM Rec	SVM F1	XGB Prec	XGB Rec	XGB F1
Piano	0.60	0.25	0.35	0.94	0.92	0.93
Accordion	0.50	0.22	0.31	0.91	0.93	0.92
Guitar ac.	0.68	0.47	0.56	0.88	**0.97**	0.92
Guitar el.	**0.95**	0.49	0.65	**0.98**	0.93	0.95
Violin	0.37	**0.82**	0.51	0.83	0.93	0.88
Cello	0.49	0.51	0.50	0.90	0.93	0.91
Contrabass	0.51	0.62	0.56	0.85	0.88	0.86
Trumpet	0.35	0.60	0.44	0.85	0.81	0.83
Trombone	0.64	0.81	**0.71**	0.87	0.91	0.89
Horn	*0.00*	*0.00*	*0.00*	0.78	0.78	0.78
Alto sax	0.38	0.08	0.13	0.92	0.63	0.75
Bassoon	0.73	0.61	0.67	0.86	0.89	0.88
Clarinet	0.50	0.22	0.31	0.95	**0.97**	**0.96**
Piccolo	0.36	0.55	0.43	0.92	0.95	0.94
Flute	0.55	0.41	0.47	0.88	0.95	0.91
Koto	0.51	0.57	0.54	0.91	**0.97**	0.94
Drums c.	*0.00*	*0.00*	*0.00*	*0.56*	*0.32*	*0.40*
Drums r.	0.10	0.05	0.07	0.59	0.67	0.63
Drums j.	0.38	0.72	0.50	0.75	0.79	0.77
45 classes	0.42	0.47	0.40	**0.83**	**0.84**	**0.83**

Table 4: Performance of the state-of-the-art SVM method compared to the XGBoost algorithm on selected instruments in RWC dataset.

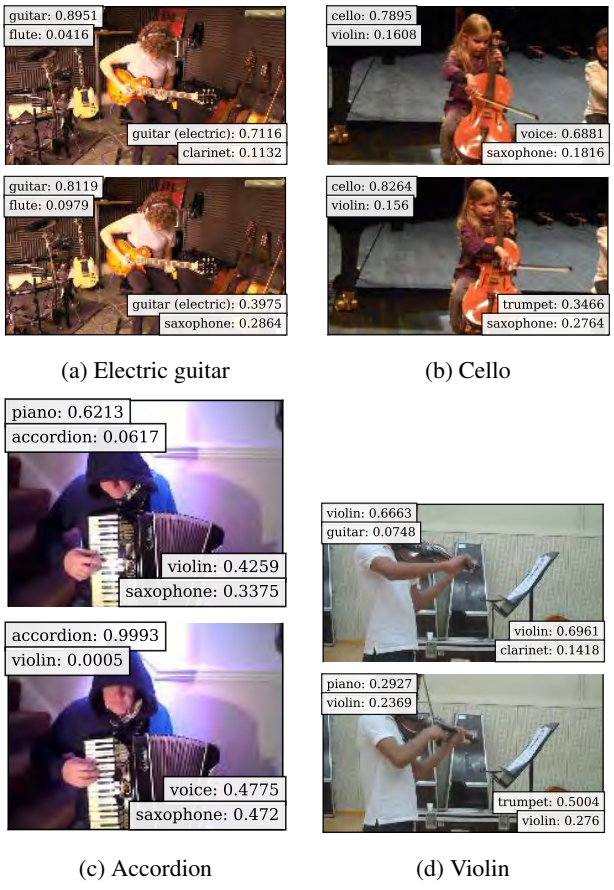
5. REFERENCES

[1] M. Schedl, E. Gómez, and J. Urbano, "Music information retrieval: Recent developments and applications," *Foundations and Trends in Information Retrieval*, vol. 8, no. 2–3, pp. 127–261, 2014.

[2] P. Herrera-Boyer, A. Klapuri, and M. Davy, "Automatic classification of pitched musical instrument sounds," in *Signal processing methods for music transcription*. Springer, 2006, pp. 163–200.

[3] B. Marenco, M. Fuentes, F. Lanzaro, M. Rocamora, and A. Gómez, "A multimodal approach for percussion music transcription from audio and video," in *Progress in Pattern Recognition, Image Analysis, Computer Vision, and Applications*. Springer, 2015, pp. 92–99.

[4] C. Liem, A. Bazzica, and A. Hanjalic, "Looking beyond sound: Unsupervised analysis of musician videos," in *Image Analysis for Multimedia Interactive Services (WIAMIS), 2013 14th International Workshop on*, July 2013, pp. 1–4.



(a) Electric guitar (b) Cello

(c) Accordion (d) Violin

Figure 4: Illustrative examples of guitar, cello, accordion and violin video frames with the top-2 prediction from audio and image sources. The best two prediction of the image-based model is located in the top left corner. The best two prediction of the audio-based model is located in the bottom right corner.

[5] A. Lim, K. Nakamura, K. Nakadai, T. Ogata, and H. G. Okuno, "Audio-visual musical instrument recognition," *73* , vol. 5, p. 9, 2011.

[6] S. Essid and G. Richard, "Fusion of Multimodal Information in Music Content Analysis," in *Multimodal Music Processing*, ser. Dagstuhl Follow-Ups, M. Müller, M. Goto, and M. Schedl, Eds. Dagstuhl, Germany: Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, 2012, vol. 3, pp. 37–52. [Online]. Available: http://drops.dagstuhl.de/opus/volltexte/2012/3465

[7] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.

[8] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei, "ImageNet Large Scale Visual Recognition Challenge," *International Journal of Computer Vision (IJCV)*, vol. 115, no. 3, pp. 211–252, 2015.

[9] J. J. Bosch, J. Janer, F. Fuhrmann, and P. Herrera, "A comparison of sound segregation techniques for pre-dominant instrument recognition in musical audio signals." in *13th International Society for Music Information Retrieval Conference (ISMIR 2012)*, 2012.

[10] F. Fuhrmann and P. Herrera, "Polyphonic instrument recognition for exploring semantic similarities in music," in *International Conference on Digital Audio Effects (DAFx)*, 2010.

[11] D. Bogdanov, N. Wack, E. Gómez, S. Gulati, P. Herrera, O. Mayor, G. Roma, J. Salamon, J. R. Zapata, and X. Serra, "Essentia: An audio analysis library for music information retrieval." Citeseer.

[12] T. Chen and C. Guestrin, "Xgboost: A scalable tree boosting system," *CoRR*, vol. abs/1603.02754, 2016. [Online]. Available: http://arxiv.org/abs/1603.02754

[13] "The university of iowa musical instrument samples (uiowa mis)," theremin.music.uiowa.edu, accessed: July 30, 2016.

[14] M. Goto, H. Hashiguchi, T. Nishimura, and R. Oka, "Rwc music database: Music genre database and musical instrument sound database." in *ISMIR*, vol. 3, 2003, pp. 229–230.

[15] G. Ye, D. Liu, I.-H. Jhuo, S.-F. Chang *et al.*, "Robust late fusion with rank minimization," in *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*. IEEE, 2012, pp. 3021–3028.

[16] T. G. Dietterich, "Approximate statistical tests for comparing supervised classification learning algorithms," *Neural computation*, vol. 10, no. 7, pp. 1895–1923, 1998.

[17] A. Karpathy, G. Toderici, S. Shetty, T. Leung, R. Sukthankar, and L. Fei-Fei, "Large-scale video classication with convolutional neural networks," in *Proceedings of International Computer Vision and Pattern Recognition (CVPR 2014)*, 2014.

[18] A. Nguyen, J. Yosinski, and J. Clune, "Deep neural networks are easily fooled: High confidence predictions for unrecognizable images," in *Computer Vision and Pattern Recognition (CVPR), 2015 IEEE Conference on*. IEEE, 2015, pp. 427–436.

[19] J. Urbano, D. Bogdanov, P. Herrera, E. Gómez, and X. Serra, "What is the Effect of Audio Quality on the Robustness of MFCCs and Chroma Features?" in *International Society for Music Information Retrieval Conference*, 2014, pp. 573–578.

MUSICAL SONIFICATION IN ELECTRONIC THERAPY AIDS FOR MOTOR-FUNCTIONAL TREATMENT – A SMARTPHONE APPROACH

Benjamin Stahl **IOhannes zmölnig**

University of Music and Performing Arts Graz - Institute of Electronic Music and Acoustics
benjamin-cosimo.stahl@student.kug.ac.at
zmoelnig@iem.at

ABSTRACT

This work presents a system which uses an *Android* smartphone to measure the wrist motion of patients in ergotherapy and creates musical sounds out of it, which can make exercises more attractive for patients. The auditory feedback is used in a bimodal context (together with visual feedback on the smartphone's display). The underlying concept is to implement a therapy aid that transports the principles of music therapy to motor-functional therapy using a classical sonification approach and to create an electronic instrument in this way. Wind chime sounds are used to sonify the patient's wrist motion, a three-dimensional parameter mapping was implemented. The system was evaluated in a qualitative pilot study with one therapist and five patients. The responses to the musical auditory feedback were different from patient to patient. Musical auditory feedback in therapy can encourage patients on one hand, on the other hand it can also be perceived disturbing or discouraging. From this observation we conclude that sound in electronic therapy aids in fields other than music therapy should be made optional. Multimodal electronic therapy aids, where sound can be toggled on and off, are possible applications.

1. INTRODUCTION

The use of electronic therapy aids in music therapy has recently gained more and more attention in research [1].

In a broader sense, music therapy can not only be applied to the treatment of psychological diseases, but also to the treatment of motor-functional impairments.

Other approaches in the treatment of such impairments are movement sonification systems. In motor-functional treatment such systems have shown to be innovative alternatives to traditional movement data monitoring systems [2, 3].

Wouldn't it be interesting to combine the field of electronic therapy aids in music therapy with the field of movement sonification in order to create an instrument-like application aiming to improve patients' physical and cognitive abilities? Fig. 1 shows how the field of research that is explored in this paper aligns between the related fields.

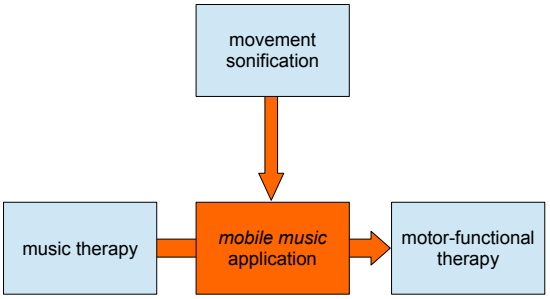


Figure 1. Position of the explored field of research with respect to other fields

Mobile music apps are applications for mobile end-user devices that incorporate sound as a central element. These applications can be virtual models of acoustic or electronic instruments as well as abstract instruments, augmented reality applications and musical games. Whatever kind a mobile music instrument be, it is often a strongly simplified version of the real model that it is based on or – in the case of an abstract instrument – (viewed from a traditional-virtuoso position) a very simple instrument. This simplicity is mostly based on the limitations of the interface (small touchscreen, no other kinaesthetic elements). Such an instrument, however, creates the opportunity of leading people with no or small musical knowledge to independently making music. The "toy character", which might be the biggest point of criticism from a traditional-virtuoso point of view, can be a great advantage for the use in a therapeutic context.

The three-dimensional accelerometer, which is nowadays integrated in all smartphones, enables the use of mobile music apps also in motor-functional therapy. Patients' movements can be sonified, which provides them an auditory feedback of their movements (possibly in addition to a visual feedback). In this way, motor-functional therapy can be more attractive and easily accessible for patients and thus encourage them to do certain exercises on their own.

We see the great potential of musical movement sonification in *mobile music apps* in therapy situations, where patients are physically as well as cognitively challenged, because both interacting with sound in a musical way and using sound as feedback address cognitive abilities. Ergotherapy (or occupational therapy) is a well suitable application.

As an application example a *mobile music* therapy aid for motor functional therapy of the wrist is developed. The

wrist is frequently treated in ergotherapy not just with orthopaedically-only impaired patients, but also with patients with neurologic impairments, in order to regain motor skills. Reestablishing motion and improving its coordination are important goals in such a therapy. These goals are pursued by repetitive and varied exercising of skills that are relevant to the respective patient's everyday life. In this connection it is an advantage if patients do exercises alone and independently.

An application such as the one described in this paper aids this form of therapy, because therapy goals are pursued in a playful way and body functions are exercised by attending an external task.

2. RELATED WORK

2.1 Electronic therapy aids in music therapy

Magee (2006) [4] describes the technical applications and constraints of electronic therapy instruments. Based on a survey the reasons for the far-reaching rejection of electronic instruments on behalf of music therapists are discussed. A lack of sufficient training possibilities for therapists was found to be the main reason of this rejection.

Magee and Burland (2008) [5] propose a treatment model for music therapy with electronic therapy aids. They name complex physical and sensory impairments, motivation problems and certain needs for self-expression as indications and cases where patients are not aware of the causal effectiveness of the treatment as contraindications for the use of electronic therapy aids.

Hadley et al. (2013) [1] give a detailed overview of both the historic context and the current situation concerning the use of music technology in music therapy.

2.2 Movement sonification

Godbout et al. (2014) [6] propose a mobile movement sonification system for athletes. They use accelerometer data from an *Armour 39* chest strap to sense the athletes motion and a sonification algorithm running on an *Android* device using *PdDroidParty* in order to provide auditory feedback to an athlete.

Vogt et al. (2009) [2] implemented a movement sonification system, where movements are sensed using an optical *VICON* motion tracking system. They use very simple mappings and metaphors in order to make their system understandable for non-specialist subjects. In physiotherapy the system can be used in order to motivate patients to do certain movements.

Pauletto and Hunt (2006) [3] present a movement sonification system for use in physiotherapy, which transfers electromyographic data to the auditory domain. A simple amplitude modulation approach was chosen for the parameter mapping. The system was evaluated in an experiment with 21 subjects. It was found that the sonification implicitly provided auditory metaphors related to nature events. The authors conclude that these metaphors contribute to the usability of the feedback tool.

3. DESIGN CRITERIA

Unlike the movement sonification systems described in section 2.2 we do not intend to use sound as an information channel in the first place in our system. Rather a musical instrument that accompanies motor-functional exercises in therapy should be created. This principal is closer to music therapy for treatment of physical inhibitions than to movement sonification in health application as it has been previously implemented. However, in order to create such a kind of instrument, we intend to use a typical parameter mapping, which is a principal coming from the field of sonification.

Since the purpose of the system is to make exercising more attractive for patients, the sound we use has to be appealing. Completely abstract sounds (such as sine tones) can not meet up with this requirement. One has to be able to create a metaphor that exalts patients' imagination. Pleasantness of the sound is also an aspect that was given weight to in the design process.

This led us to the idea of building a wind chime instrument model. In this model we intended to create several unique sounds with different characteristics that are controlled by the patient's hand position in order to make the application as intuitive as possible. Furthermore, nature metaphors should help the patient and the therapist recognize characteristic sounds. These metaphors are described in section 4.5.

4. MOTOR-FUNCTIONAL WRIST TREATMENT - SYSTEM DESCRIPTION

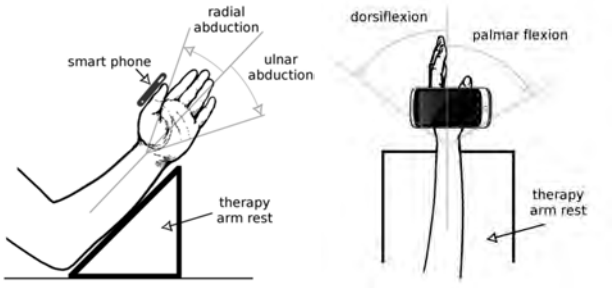
The system we implemented uses an *Android* smartphone, a special glove which is used to fix the smartphone to the patients hand and a special therapy arm rest to stabilze the patient's lower arm. Fig. 2 shows the additional components which are required besides the smartphone. An instrument app which provides visual and musical auditory feedback to a patient was implemented on the Android smartphone. For visual rendering, the *Processing* Android library and the Processing GUI library *controlP5* was used. Sound synthesis is done with *Pure Data* using the *libpd* [7] Android API to integrate the patch in the Android app. The sound is played back via the smartphone's loudspeaker.

(a) Therapy arm rest

(b) Special glove with vacuum cup to attach the smartphone to the patients hand

Figure 2. Additional system components

Figure 3. System in use



(a) viewed from inner side (b) viewed from top

Figure 4. Position of the (left) lower arm when using the therapy aid; illustration of the wrist motion

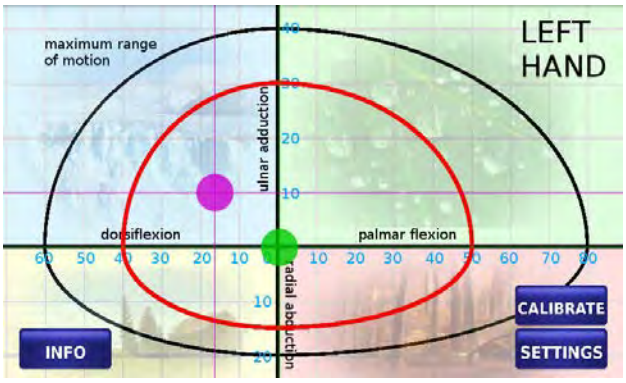


Figure 5. Graphical surface of the app

4.1 Capturing the motion

The motion of the human wrist can be descibed as a rotation of the hand around two axes. The palmar-dorsal movement is a flexion towards the palm of the hand (*palmar flexion*, maximum angle: 80°) respectively an extension towards the back of the hand (*dorsiflexion*, maximum angle: 60°). The radial-ulnar movement describes the moves of the hand towards the little finger (*ulnar abduction*, maximum angle: 40°) and towards the thumb (*radial abduction*, maximum angle: 20°) [8].

Movements of the patient's wrist are indirectly measured using the smartphone's accelerometer; we decided to use the following approach for the detection of the hand position: In therapy the thumb side of the hand, on which the smartphone is attached in landscape format (see Fig. 3 and 4), points towards the patient. The patient's lower arm lies on a special therapy arm rest. The angle between the arm rest and the horizontal is at least 45°.

When viewing the smartphone's rotation in nautical angels *yaw*, *pitch* and *roll*, a palmar-dorsal movement of the wrist corresponds to a *pitch* movement. A radial-ulnar movement analogously corresponds to a *roll* movement (where the roll rotation axis is the y-axis of the smartphone). This consideration requires, that no supination (outwards rotation) or pronation (inwards rotation) of the lower arm is performed.

Now, when assuming that no other acceleration but gravity affects the smartphone and neither the pitch nor the roll rotation axis points in the direction of the gravitation vector, the pitch and the roll angle can be calculated from the accelerometer data.

When the palmar-dorsal rotation axis is parallel to the gravitation vector (roll angle is close to +/- 90°), palmar-dorsal movements can not be properly captured anymore. For this reason, the smartphone must never be horizontally oriented. The minimum 45° tilt of the lower arm follows from a maximum ulnar abduction of 40° and a safety buffer of 5°. Depending on the smartphone's roll angle in the initial position (arm lies on the therapy rest, no radial or ulnar movement), a calibration is executed for the calculation of the radial-ulnar wrist abduction.

The wrist position which is detected in this way is denoted as

- $PD(t)$ ... palmar flexion (negative values) or dorsiflexion (positive values) (range: -60° ... 80°)
- $RU(t)$ ... radial abduction (negative values) or ulnar abduction (positive values) (range: -20° ... 40°)

4.2 Data processing and visual feedback

Fig. 5 shows the graphical surface of the app. The transparent background images in the particular quadrants of the graphical surface are explained below in section 4.5.

The range of motion of an healthy human wrist can be described as an ellipse in a coordinate system which is defined by the two motion angles (palmar-dorsal; radial-ulnar) [9]. Therefore, the maximum (healthy) range of motion is shown as a black ellipse on the graphical surface. The exercise range of motion, which can be set by the therapist, is shown as a red ellipse, which is distorted in the following way: In each half plane, the ellipse for the maximum range of motion is scaled to the maximum exercise angle set by the therapist in order to obtain the exercise range of motion (upper half plane - ulnar abduction $UA_{ex_{max}}$, lower half plane - radial abduction $RA_{ex_{max}}$, inner half plane - palmar flection $PF_{ex_{max}}$, outer half plane - dorsiflexion $DF_{ex_{max}}$).

A green dot on the screen represents the current position of the patient's hand ($PD(t)$, $RU(t)$). An optional violet dot serves as a guiding point, as it performs movements (random movement or circular movement) and the patient should follow it with the green dot. The guiding point only moves inside the boundaries of the set exercise range of motion.

From the detected hand position, also the following quantities are calculated:

$$v_{wrist}(t) = \sqrt{\dot{PD}^2(t) + \dot{RU}^2(t)}$$

$$\ldots \text{ absolute angular velocity; unit: } \left[\frac{^\circ}{s}\right] \quad (1)$$

$$a_{wrist}(t) = \dot{v}_{wrist}(t)$$

$$\ldots \text{ absolute angular acceleration; unit: } \left[\frac{^\circ}{s^2}\right] \quad (2)$$

As patients should not make their movements too edgy, a_{wrist} should not become too high. We therefore implemented a multimodal warning clue, when a_{wrist} exceeds a certain value. The threshold can be manually changed. On the visual side, a short red flashing of the screen signalizes a movement being to edgy. The auditory supplement of this clue is described in section 4.3

4.3 Musical sonification

In order to create a musical sonification of the wrist motion, an instrument model of a wind chime was built. A sample of a wooden chime sound and a sample of metallic chime sound are used to create two different instances of the wind chime. The playback speed of the samples is varied in order to create different musical pitches. We use discrete pitches on a pentatonic scale. The samples were polyphonically played back, each playback instance starting at a random point of time. The density and the level of playback as well as the pitch range are parameters that can be varied to change the sound. Furthermore, the balance between the wooden instance and the metallic instance of the wind chime can be varied.

The following mappings were used for the musical sonification:

1. Angular velocity → density, level

The more the wrist is moved, the denser and louder the sound becomes. Fig. 6 illustrates the mapping. For the computation of the time distance between two note onsets, first a probability value p is calculated from the angular velocity as

$$p = \min(v_{wrist} \cdot 0.017 \frac{1}{\frac{^\circ}{s}}, 0.5). \quad (3)$$

For practical reasons, a clock with a fixed clock cycle interval of 4ms creates discrete time events. For each event, a uniform distributed random value between 0 and 1 is computed. If the value is smaller than p, a chime sample is played back, otherwise nothing is done. This results in the fact that each 4ms a chime sample is played back with the probability p. The temporal resolution of 4ms is adequate for this synthesis: The user can not tell that the events are actually triggered by a clock with a fixed cycle interval, and the CPU load is compatible. The mapping of the angular velocity v_{wrist} to the probability p is linear, using the probability factor $0.017\frac{s}{^\circ}$, which was subjectively adjusted, so the system reacts already to relatively small movements, but natural movements do not immediately lead to an extremely dense sound. Furthermore, p is clipped to 0.5 (this is equivalent to an angular velocity of 29.4 $\frac{^\circ}{s}$) in order not to let the sound get to dense and to keep some randomness even with fast movements.

The time interval between two note onsets can therefore be calculated as

$$\tau_d = 4\,ms \cdot Q, \quad (4)$$

where Q is a negative binomial distributed random variable:

$$Q \sim \mathcal{NB}(1, p). \quad (5)$$

The resulting average time distances thus reach from 8 ms to ∞.

Figure 6. Schematic representation of density and level mapping; please note that the illustrated notes and breaks are not literally notes with the respective note values, the time distance between the notes is (partially randomly) changing

The angular velocity also modifies the amplitude level. The relative level factor is calculated as

$$l = \left(v_{wrist} \cdot 0.34 \frac{1}{\frac{^\circ}{s}} + R\right) dB, \quad (6)$$

where R is a uniformly distributed random variable with

$$R \sim \mathcal{U}(-20, 20) \quad (7)$$

The proportionality factor $0.34\frac{s}{^\circ}$ in this parameter mapping was also subjectively adjusted, so the level changes in conjunction with the density changes result in a consistent dynamic response.

2. Palmar/dorsal rotation → wood/metal balance

Two instances of the wind chime were created, a wooden one and a metallic one. Depending on the palmar/dorsal rotation of the wrist, a crossfade is performed between the two instances. Therefore, the palmar/dorsal deviation is normalized according to the set exercise range of motion:

$$PD_{norm}(t) = \frac{PD(t) + PF_{ex_{max}}}{DF_{ex_{max}} + PF_{ex_{max}}} \quad (range: 0 \ldots 1) \quad (8)$$

From this value, the levels of the wood and the metal chimes are calculated according to the crossfade function displayed in Fig. 7. We decided to create a rather steep crossfade in order to make changes clearly noticeable. Therefore, the amplitudes were clipped at $PD_{norm}(t) = 0.3$ resp. $PD_{norm}(t) = 0.7$. For $0.3 < PD_{norm}(t) < 0.7$, also very small changes in the palmar/dorsal rotation can thus be noticed in the sound.

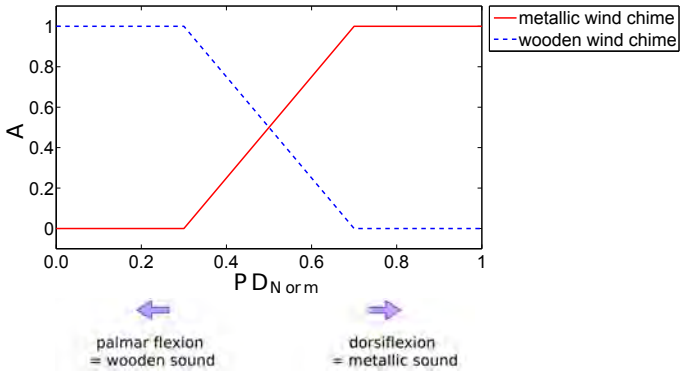


Figure 7. Schematic representation of balance mapping

3. Radial/ulnar deviation → pitch

The tonal repertoire that is used by the wind chimes at one certain point of time is a four-note section from a pentatonic scale. For one particular note at this point of time, the pitch (= playback speed) is randomly selected from the four possible pitches. Choosing from four different pitches should ensure, that the resulting polyphonic chime sound is rich in variety and thus pleasant. The tonal repertoire for one point of time could not be chosen larger, because then the mapping, which is described below, would not create enough distinguishable tonal repertoires any more.

Analogously to the normalization of the palmar/dorsal rotation, the radial/ulnar rotation is normalized:

$$RU_{norm}(t) = \frac{RU(t) + RA_{ex_{max}}}{UA_{ex_{max}} + RA_{ex_{max}}} \quad (range : 0 \ldots 1) \tag{9}$$

We mapped this normalized measure to the pitch of the selected section of a C-major pentatonic scale reaching from $C3$ up to $D7$. The mapping was performed in such a way, that for $RU_{norm}(t) = 0$, the lowest section of the scale was the tone repertoire $(C3, D3, E3, G3)$ and for $RU_{norm}(t) = 1$, the highest section was the tone repertoire $(G6, A6, C7, D7)$. Fig. 8 schematically visualizes this mapping.

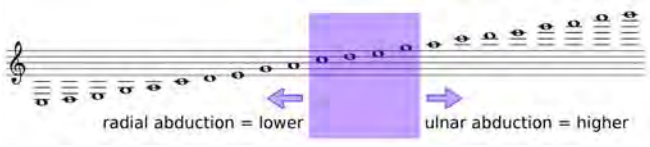


Figure 8. Schematic representation of pitch mapping

The auditory supplement of the visual warning clue described in the previous section is a loud drumbeat which is played back while the screen flashes red.

4.4 Unguided / guided exercises

Different usage modes can be set. In the basic mode, the violet dot which serves as a guiding point for the patient's movements was not displayed. In this mode, the instrument character of the app is pointed out most clearly. Patients don't have a certain exercise, therefore their main focus is on creating interesting sounds. With the guiding point visible, there exist three guiding modes. In all modes, the guiding point's speed depends on the angular distance of the patient's hand position to the guided position: When the patient's hand position is close to the guided position, the guiding point moves faster, when it is further away, it moves slower. This ensures that patients do not rush to follow the guiding point, because it is much more important, that the movements are smooth, than making them fast.

The first guiding mode is a random ribbon movement. Therefore, the palmar-dorsal and the radial-ulnar coordinates change over time as sine functions, which are asynchronous (different frequencies). The constantly changing phase shift creates a ribbon movement, that seems random.

The second guiding mode is a clockwise circular movement (circumduction). Therefore, the palmar-dorsal and the radial-ulnar coordinates change over time with synchronous sine functions, one being 90° phase shifted with respect to the other. The radius is also varied over time.

The third guiding mode is a counter-clockwise circular movement, it is created analogously to the clockwise circular movement.

The guiding point coordinates are scaled and cropped in order to remain within the set exercise range of motion.

4.5 Nature metaphors

In order to make the mapping easier to understand, metaphoric nature images are assigned to the sound characteristics. As seen in Fig. 5, photos of landscapes are transparently displayed on the screen to remind patients of the metaphors. The metaphor for high-pitched wooden sounds are raindrops, because high pitched percussive sounds with a short sustain resemble the sound of rain. Low-pitched wooden sounds resemble the sound of colliding wood blocks, therefore the metaphor for this characteristic is a forest. The metaphor for a metallic, high sounds is a glacier, high pitched metallic chime sounds can be associated to ice crystals. The metaphor for low-pitched metallic sounds is a field at an alp, because the sound resembles cowbells.

5. EVALUATION

The system was evaluated in a pilot test with one therapist and five patients. Both the therapist and the patients were asked to fill in questionnaires after using the system in therapy. The therapist was extensively informed about how to use the system both orally and written. The patients were informed about the intention of the study in written form. Written consent was acquired from both patients and therapist.

5.1 Research questions and hypotheses

With the pilot study, we aimed to find answers to the following research questions:

- *Could a system that is intuitively understandable be successfully created?*

- *How do patients and therapists estimate the role of the sound in the application?*
- *Does the system motivate patients to do more exercises?*

We hypothesized that the system is intuitive due to the multimodal feedback: Even if one (the auditory or the visual) feedback is not perceived intuitive enough and therefore would lead to confusion if presented solely, the other will compensate and help creating a comprehensible multimodal perception. Also the nature metaphors we gave the therapist and patients as an orientation help should improve the intuitiveness of the system. We did not make any hypothesis about the role of sound that the patient sees in the application, it was clear that (as in most cases with auditory display) pleasantness might be a problematic aspect. Another hypothesis was that for patients who don't perceive the sound disturbing, such a novel therapy aid would be interesting and therefore motivate them to do more exercises.

5.2 Participant demographics

Five ergotherapy patients took part in the study. The patients were aged between 57 and 83 years, four of the five subjects were female, one was male. The patients were all treated for neurological diseases. Two of the five subjects suffered from multiple sclerosis, three received treatment because they have had a cerebrovascular accident (stroke).

5.3 Conduct

The application of the therapy aid was integrated in a normal therapy session. With four of the patients the therapy session was held in a home visit, with one the therapy session was held in a practice room. The smartphone used in the evaluation was a *Samsung Galaxy S3 mini*. Before exercising with a patient, the therapist set the exercise range of motion in the app settings according to the patient's abilities.

First, the patients used the system without having the exercise of following a guiding point. They should freely try out, how sound characteristics changed, when moving the hand in different directions. The sound metaphors and the nature images on the screen were explained in order enhance this learning process. The patients were also instructed to move their hand as smoothly as possible, the warning clue was demonstrated.

When the patients had a feeling for the changes in sound they could create, the therapist set different guiding modes (random movement, clockwise circumduction, counterclockwise circumduction). The patients were given the exercise of following the (violet) guiding point, without making movements that are too edgy.

The whole application of the therapy app took a total of approximately 20 minutes. Filling in the questionnaire took approximately 5-10 minutes.

5.4 Methods, results and discussion

First, both patients and therapist were asked about their previous experience with electronic therapy aids, their musical knowledge / experience and if they generally find the use of a smartphone in therapy sensible. Therapist and patients had no or few experience with electronic therapy aids, considered their musical knowledge rather small and – on average – found the use of a smartphone in therapy rather sensible.

The questionnaire which was used to evaluate the system featured several statements. The patients should rate on a five-point Likert scale [10], to which extent the particular statements applied to them. The Likert scale points were: *does not apply at all*, *rather does not apply*, *yes and no*, *rather applies* and *totally applies* Furthermore, the questionnaire featured some open-ended questions, where patients should elaborate different aspects of their experiences with the system. A similar questionnaire was created for the therapist.

Most of the patients and the therapist considered the system rather intuitive. All of the patients found the statement *"I clearly understood my exercises."* totally applicable. Focusing on the exercises was also not considered a problem, patients either rated the statement *"I could focus well on the exercises."* with *rather applies* or *totally applies*. The mappings in the motion sonification (statement *"I quickly and clearly understood how my motion affected the sound."* was considered rather intuitive by four of the five patients. The visual feedback was also perceived intuitive. These findings confirm the hypothesis we made about the intuitiveness of the system. In contrary to our hypotheses, the responses to the nature metaphor images on the screen were rather negative, 4 of the 5 patients totally disagreed to the statement *"I find the nature metaphors and the landscape images on the screen useful"*. We suppose that the negative response is due to a non-optimal integration of the images in the graphical surface. The images were to small to view them on a smartphone screen and could therefore not be identified correctly. The therapist however reported that when explaining the application to her patients, the nature metaphors were very useful and might enhance the whole sound perception. Therefore we do not reject the hypothesis that the nature metaphors improve the intuitiveness of the system. The audiovisual warning clue in case of too edgy movements was also not found to be very sensible. The agreement to the statement *"I find the drumbeat, which was played back when movements were too edgy, useful."* was pretty diverse. Two of the subjects rather disagreed, one fully disagreed, one was undecided and one rather agreed. However, the therapist fully agreed to the statement. This shows that warning clues seem to provide good feedback but - from a psychological point of view - are rather inappropriate in therapy, because they might discourage patients.

Concerning the role of the sound in the application the following results were found: The sound was considered moderately pleasant, where some patients rather found to the statement *"I perceive the sound as pleasant."* applicable and others did not. When we asked patients concerning their agreement to the the statement *"I would have preferred to exercise without sound."*, two of them fully agreed, one was undecided, one rather disagreed and one fully disagreed. When asked if they had the feeling of bumping a real wind

chime, most of the patients confirmed, that the simulation of a real wind chime worked for them. We may infer that patients' responses to the use of sound in therapy are very diverse. Some patients could profit from the multi-modality of the application, others could not.

Also concerning the research question if the system would encourage patients to do more exercises the evaluation results were pretty diverse. The patients' responses to the statement *"With the app, I would exercise more by myself than I used to before."* were widely spread (with a tendency that patients rather would not do more exercises than they used to before). It could be seen that patients who would have preferred to exercise without sound rather tended to not finding this statement applicable while at least some of the others were partially willing to do more exercises at home with the app. The therapist reported that the responses to this statement were also strongly correlated with the patients' basic attitude in exercising on their own: Patients that already did their exercises frequently found that the application would not help them in exercising even more.

Although we can not show that our system motivates patients to do more exercises, when asked about what they found good about the system, patients who were not disturbed by the sound, answered that they liked getting to know and learning something new in therapy.

6. CONCLUSIONS AND OUTLOOK

The evaluation of the application example for motor functional therapy of the wrist showed that the perception of the musical sonification in the electronic therapy aid was very different for individual patients. A very intuitive application could be created. This is also confirmed by the fact that all patients clearly understood their exercises and could comprehend the parameter mapping despite complex neurological impairments, no or few experience with electronic therapy aids and small musical knowledge. Nevertheless, the sound in the application was perceived disturbing instead of motivating by some patients. This indicates that sound in electronic therapy aids should be made optional in fields other than music therapy. As we did not design the sound as the main carrier of information in our application, it can theoretically also be used without sound if desired. We infer that multimodal applications such as the proposed system are a well-suitable application for musical sonification in motor-functional treatment. It needs to said that due to the small number of subjects in the evaluation the findings are not based on any statistical significance in the results, but on trends we observed in qualitatively viewing the results.

In a new evaluation the research question if the system motivates patients to do more exercises should be divided in several aspects in order to find a differentiated answer to this research question.

Answers to the open-ended questions showed that some aspects of the implementation of the system still have to be improved: the graphical surface could be simplified by removing the (probably unnecessary) landscape pictures. Feedback about how smooth the patient's movements are should be given in a more encouraging way than in the proposed system. Adding a game-like structure (levels, rewards, ...) with an appropriate auditory supplement (e.g. auditory icons or earcons) could also improve patients' motivation to exercise.

After improving the usability of the systems, musical sonification for movements of other joints could be implemented in a similar way. Also different instruments schemes (from which the patient/therapist can freely choose) could be added in order to create a more varying user experience.

Acknowledgments

We would like to acknowledge the contribution of the ergotherapist Nicole Stahl, who helped designing the therapy aid and conducted the evaluation. Furthermore, we would like to thank the ergotherapy centers *Praxis für Ergotherapie Martina Dengler*, Jettingen, Germany and *Praxis für Ergotherapie & Craniosacraltherapie Birgit Stähle*, Bondorf, Germany for enabling the evaluation.

7. REFERENCES

[1] S. Hadley *et al.*, "Setting the scene," in *Music technology in therapeutic and health settings*, W. L. Magee, Ed. *London:* Jessica kingsley publishers, 2013.

[2] K. Vogt *et al.*, "Physiosonic - movement sonification as auditory feedback," in *Proceedings of the 15th International Conference on Auditory Display*, 2009.

[3] S. Pauletto and A. Hunt, "The sonification of emg data," in *Proceedings of the 12th International Conference on Auditory Display*, 2006.

[4] W. L. Magee, "Electronic technologies in clinical music therapy: A survey of practice and attitudes," *Technology and Disability*, vol. 18, no. 3, pp. 139–146, 2006.

[5] W. L. Magee and K. Burland, "An exploratory study of the use of electronic music technologies in clinical music therapy," *Nordic Journal of Music Therapy*, vol. 17, no. 2, pp. 124–141, 2008.

[6] A. Godbout, C. Thornton, and J. Boyd, "Mobile sonification for athletes:a case study in commercialization of sonification," in *Proceedings of the 20th International Conference on Auditory Display*, 2014.

[7] P. Brinkmann *et al.*, "Embedding pure data with libpd," in *Proceedings of the Pure Data Convention 2011*, 2011.

[8] G. Aumüller et al., *Duale Reihe Anatomie*. *Stuttgart:* Thieme, 2014.

[9] P. Salvia *et al.*, "Analysis of helical axes, pivot and envelope in active wrist circumduction," *Clinical Biomechanics*, vol. 15, no. 2, pp. 103–111, 2000.

[10] R. Likert, "A technique for the measurement of attitudes," *Archives of Psychology*, vol. 140, pp. 5–55, 1932.

SOUNDSCAPE PREFERENCE RATING USING SEMANTIC DIFFERENTIAL PAIRS AND THE SELF-ASSESSMENT MANIKIN

Francis Stevens
AudioLab,
Department of Electronics,
University of York, UK
fs598@york.ac.uk

Damian T Murphy
AudioLab,
Department of Electronics,
University of York, UK
damian.murphy@york.ac.uk

Stephen L Smith
Intellignet Systems Group,
Department of Electronics,
University of York, UK
stephen.smith@york.ac.uk

ABSTRACT

This paper presents the findings of a soundscape preference rating study designed to assess the suitability of the self-assessment manikin (SAM) for measuring an individual's subjective response to a soundscape. The use of semantic differential (SD) pairs for this purpose is a well established method, but one that can be quite time consuming and not immediately intuitive to the non-expert. The SAM is a questionnaire tool designed for the measurement of emotional response to a given stimulus. Whilst the SAM has seen some limited use in a soundscape context, it has yet to be explicitly compared to the established SD pairs methodology. This study makes use of B-format soundscape recordings, made at a range of locations including rural, suburban, and urban environments, presented to test participants over a 16-speaker surround-sound listening setup. Each recording was rated using the SAM and set of SD pairs chosen following a survey of previous studies. Results show the SAM to be a suitable method for the evaluation of soundscapes that is more intuitive and less time-consuming than SD pairs.

1. INTRODUCTION

Environmental noise has been increasingly recognised as a form of pollution equally important as more traditional pollutants [1]. In order to understand how the negative effects of noise pollution arise, a method extending beyond noise level measurement is required. One such method is auralisation where a measured or simulated soundscape can be presented in a lab environment, and subjective responses to that soundscape can then be measured [2].

The established method for gathering these subjective responses is the use of SD pairs [3] to rate soundscapes in terms of multiple scales. This method can be time consuming, due to the cumbersomeness of measuring perhaps 18 or more ratings for each stimulus in a given test. It can also be non-intuitive for non-experts where specific terms are used, and relies on an understanding of the terms used which requires translation and validation for use in multiple languages [4]. The SAM consists of only three scales presented in pictorial form and was generated in an attempt to design a preference rating tool free from these problems.

It is hypothesised that a comparison of soundscape preference rating results between a set of SD pairs and the SAM will show the SAM to be a directly comparable and equally useful tool for the analysis of subjective soundscape experience.

This paper first considers the test methodology, including the decisions made in data collection and the subjective assessment methods used. The results from this experiment are shown, with the SAM and SD pair results compared using correlational analysis. The ratings of the recorded soundscapes are also presented in terms of Russell's circumplex model of affect [5]. This paper ends with a concluding section showing the above hypothesis to be supported by the collected evidence. This section also considers avenues for further research.

2. METHODOLOGY

2.1 Data Collection

During Summer 2015 data were collected from 8 locations around the north of England covering a wide range of environments from rural to suburban and urban. At each location audio-visual recordings were made using a 4-channel Soundfield surround-sound microphone and 6-GoPro cameras mounted on a cube allowing for the capture of spherical images. The visual data were collected for use in further experiments. The aim when choosing the recording locations was to cover as wide a range of sound sources, noise levels, and visual features as possible.

2.1.1 Sound Sources

In order to select a set of recording locations covering as wide a range of soundscapes as possible, previously identified categories of soundscapes and their components sound sources had to be considered. In a significant quantity of soundscape research [6–11] three main groups of sounds are identified:

- **Natural sounds:** These include animal sounds (bird song is an oft-cited example), and other naturally occurring environmental sound.
- **Human sounds:** Any sounds that are representative of human presence/activity that do not also represent

industrial activity. Such sounds include footsteps, speech, coughing, laughter etc.

- **Industrial sounds:** Mechanical sounds, such as traffic noise, activity on a building site, or aeroplane noise.

The purpose of covering such a wide range of sources was to ultimately generate a set of stimuli that will elicit a wide range of emotional responses. Generally speaking, natural sounds are the most preferred, human sounds are given a neutral rating, and mechanical/industrial sounds are disliked [10].

2.1.2 Recording Duration

Several minutes of material were recorded at each location, from which two 30-second long clips were extracted. Table 1 contains details of the sound sources present in the two 30-second long clips chosen to represent each recording location. Where here the clips are numbered as '1' and '2' for each location, when referred to more generally they have each been given a number between 1 and 16, determined by the order of the locations. For example, the clips numbered 3 and 4 are respectively the 1^{st} and 2^{nd} clips recorded at location 2.

A survey of previous literature showed the duration of recordings used for soundscape reproduction to vary considerably. Whilst Harriet made use of 7-minute long soundscape representations [8] constructed artificially from recorded material, other studies typically use shorter recordings (especially those presenting visual and aural stimuli simultaneously). For example both Anderson et al. [10] and Viollon et al. [7] used 20-second long recordings. Pheasant et al. have used 32-second long recordings [12, 13], and both Watts and Pheasant and Gifford and Ng make use of recordings lasting 1-minute [11, 14]. Axelsson's work as part of the Sound Cities project used binaural recording of 46-seconds in length, presented with a set of six still images of the recording site [15]. Rummukainen et al. used even shorter recordings only 15-seconds in length [16].

The majority of these have considered visual stimulus alongside aural information, and most audio only studies have made use of soundwalks [3, 8, 17–19] which are naturally longer in duration. Work by Fröhlich et al [20] has previously confirmed the ecological validity [21] of short (c. 10-second long) video clips in quality of experience studies. Following this survey it was decided that 30-second long clips would be suitably long to present an immersive and ecological valid scenario to test participants without making the test too long.

2.2 Subjective Assessment

This section covers the methods of subjective assessment to be used in rating the recorded soundscapes. The purpose of this experiment was to assess the suitability of the Self-Assessment Manikin for soundscape preference rating by comparing results gathered from the use of the SAM with those obtained from ratings made with a set of semantic differential pairs.

2.2.1 Semantic Descriptors

The us of SD pairs is a method originally developed by Osgood to indirectly measure a person's interpretation of the meaning of certain words [22]. The method involves the use of a set of bipolar (typically 7-point [23]) descriptor scales, for example 'Weak - Strong', allowing the user to rate a given stimulus. Factor analysis can then be performed on the results to determine underlying patterns connecting the various descriptor pairs [4] .

The use of Semantic Differential (SD) pairs for the assessment of soundscape quality is well established [3,8,18, 23–27], and includes the use of both connotative and denotative scales. Denotative scales relate to the acoustic or psychoacoustic properties of the soundscape, whereas connotative scales measure the emotional meaning [26]. Table 2 shows the set of SD pairs used in this experiment. It was generated from a survey of previous studies, and the table shows where each of the SD pairs selected have been used in other studies.

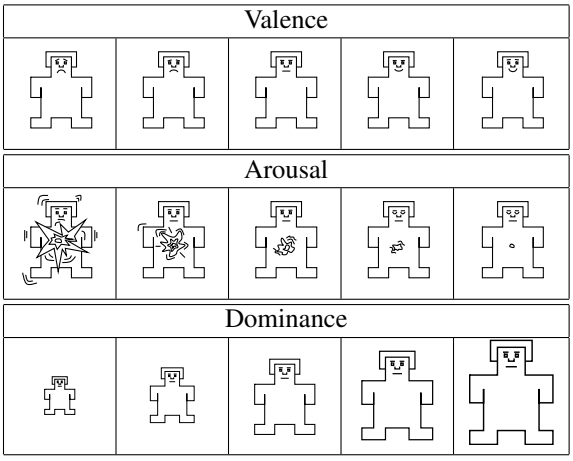


Figure 1. The Self-Assessment Manikin (SAM) as used in this experiment, after [4].

2.2.2 The Self-Assessment Manikin

The Self-Assessment Manikin (SAM) is a method for measuring emotional responses developed by Bradley and Lang in 1994 [4]. It was developed from factor analysis of a set of SD pairs rating both aural [28, 29] and visual stimuli [30] (using both the International Affective Digital Sounds database, or IADS, and the International Affective Picture System, or IAPS) . The three factors developed for rating emotional response to a given stimuli are:

- **Valence:** How positive or negative the emotion is, ranging from unpleasant feelings to pleasant feelings of happiness.
- **Arousal:** How excited or apathetic the emotion is, ranging from sleepiness or boredom to frantic excitement
- **Dominance:** The extent to which the emotion makes the subject feel they are in control of the situation,

Site	Clip 1 Sound Sources	Clip 2 Sound Sources
1. Dalby Forest	Birdsong, owl hoots, wind	Birdsong, goose honking, insects, aeroplane noise
2. Dalby Forest Lake	Wind, birdsong, insects, single car	Wind, birdsong, insects, water splashing
3. Hole of Horcum	Birdsong, traffic, bleating	Birdsong, traffic, conversation
4. Fox & Rabbit Inn	Traffic, car door closing	Traffic, car starting, footsteps
5. Smiddy Hill	Car starting, car door closing, traffic	Traffic, birdsong
6. Albion Street	Busker performance, footsteps, conversation, distant traffic	Workmen, footsteps, conversation
7. Park Row	Traffic, buses, wind, 'flute' playing	See clip 1 details
8. Park Square	Conversation, traffic, birdsong, shouting	Workmen, conversation, birdsong, traffic

Table 1. Details of the sound sources present in the two 30 second long clips used for each location.

#	Semantic Differential Pair	Harriet [8]	Kang [3]	Davies [24]	Viollon [25]
1	Quiet-Noisy	×	×	×	×
2	Comfort-Discomfort	×	×	×	×
3	Unique-Common (Interesting-Boring)	×	(×)	(×)	(×)
4	Monotonous-Varied (Varied-Simple) [Static-Changing]	×	(×)	(×)	[×]
5	Pleasant-Unpleasant	×	×	×	
6	Harmonious-Disharmonious (Gentle-Harsh)	×	(×)	(×)	
7	Soft-Rough (Soft-Hard)	×	(×)	(×)	
8	Natural-Artificial (Rural-Urban)		×	×	(×)
9	Social-Unsocial (Friendly-Unfriendly)		×	×	(×)
10	Calming-Agitating		×	×	×
11	Meaningful-Meaningless (Informative-Uninformative)		×	(×)	

Table 2. Details of the SD pairs used in this study, with their use in previous studies indicated.

ranging from not at all in control to totally in control.

These results were then used by Bradley and Lang to create the SAM itself as a set of pictorial representations of the three identified factors. The version of the SAM used in this experiment is shown in Figure 1.

The SAM has been used a select number of times for soundscape analysis, recently by Watts and Pheasant [11], and combined with concurrent physiological measures by Hume and Ahtamad [31]. However, a direct comparison of SD pair ratings with SAM results has not been conducted. Other studies have investigated the use of Russell's circumplex affect model [5] to study urban environments. Hull and Harvey found a relationship between the physical characteristics of suburban parks and affective states: tree density, and the presence of undergrowth and pathways [32], while Hanyu found that green, open, and well-kept spaces are related to positive valence, but that the presence of 'disorder elements' (vehicles, wires) is related to negative affective response [33]. Also of note is Viollon and Lavandier's identification of valence and arousal as the two main underlying factors in assessment of environmental quality [25], indicating strong possibility of a high correlation between SD pairs and the SAM.

2.3 Test Procedure

Each test subject is first presented with the pre-experiment statement and consent form, followed by a demographic questionnaire and a preview of the subjective assessment questionnaire they will use to rate each presented soundscape. This was to give them the opportunity to raise any questions they may have about the questions they will be answering, and to familiarise themselves with the test procedure. Following this they are lead into the listening space, as shown in Fig. 2.

Figure 2. Test participant in the listening space.

Subjects are then presented with each of the soundscape recordings in random order over the 16-speaker surround-sound listening setup. After each recording has finished

they are given time to fill out a subjective assessment form for each one. The typical duration of the entire procedure was around 40 minutes.

All of the questionnaire forms were prepared for presentation online, with only the pre-experiment statement and consent form, and a sheet of the term definitions presented as a hard copy [1] .

3. RESULTS

3.1 Correlation Analysis

In order to compare the results for each rating scale with one another, the results for each scale were first normalised for test subject by calculating z-scores [34]. The correlation of each rating scale with each other one was then calculated for each subject (according to Pearson's R [35]), and then the mean correlation across the subjects was calculated. This mean was then compared with the calculated critical r value, and then plotted according to its significance and whether the correlation found was positive or negative. This critical value is given by

$$r_{\mathrm{Critical}} = \frac{\mathtt{TINV}(1-\frac{\alpha}{2},\mathrm{df})}{\sqrt{(\mathtt{TINV}(1-\frac{\alpha}{2},\mathrm{df}))^2+\mathrm{df}}} \qquad (1)$$

where r_{Critical} is the minimum critical correlation value, TINV computes the inverse of the cumulative distribution function of the t-distribution, using the degrees of freedom df, and the critical probability value α [36]. For these results $\mathrm{df} = 14$ (i.e. $n-2$), and $\alpha = 0.05$, giving an r_{Critical} value of 0.4973 [36].

Figure 3 shows the results of this significance testing for each pair of rating scales. White square represent no significant correlation (the white squares with black crosses indicate where the correlation value is for a rating scale's correlation with itself i.e. $r = 1$). The red squares indicate positive correlation, and the blue squares indicate negative correlation. For the SD pairs the direction of the correlation is given where the second descriptor is positive, and the first descriptor is negative. For example, the negative correlation between Valence and the Quiet-Noisy SD pair indicates a significant correlation between increased Valence rating and Quiet-Noisy ratings closer to the Quiet end of that scale.

3.1.1 Uncorrelated Scales

There are several features of the collected data indicated by Figure 3 that merit discussion. One is that the Social-Unsocial, Informative-Meaningless, and Immersion scales are not correlated significantly with any rating scales. In the case of the Immersion scale this is a positive result, as it indicates that for a set of recordings all presented in the same format are similarly immersive independent of context.

[1] One test participant was presented with a pen-and-paper version of the survey instead, due to their lack of comfort in using the computerised version. This was considered to be suitable for inclusion after Bradley and Lang's findings of correlation values of 0.99, 0.94, and 0.79 for each dimension of the SAM (valence, arousal, and dominance respectively) compared between a pen-and-paper version and a computerised version [4].

This result for the Social-Unsocial and Informative-Meaningless scales is a reflection of some informal feedback from test participants indicating a perceived ambiguity in these two scales. In the case of Social-Unsocial there is a certain paradox present where an ostensibly sociable environment (e.g. Location 6 Albion St, Leeds) sounds overly busy and does not feel like an inviting place to take part in social activities: a fact evidenced by the different ratings for this scale for the two clips recorded at this location. Conversely, a quiet or empty soundscape (e.g. Location 1 Dalby Forest) may sound like an encouraging place for an activity to take place, even if the soundscape itself does not contain any social sounds.

A similar confusion is evident in the results for the Informative-Meaningless scale, as there is no explication of what constitutes 'true' meaning and information. For example, traffic noise conveys information regarding the rate and volume of passing cars present in the soundscape but is in many other ways meaningless. This maybe indicates a blurred boundary between the 'keynote' and 'signal' sounds as identified by Schafer [37]. Without a wider context (such as would be provided by a presented visual setting), it is not facile to identify which sounds provide a backdrop, and those sound which comprise the foreground of the soundscape (if indeed there are any such sounds present).

The Interesting-Boring and Varied-Monotonous scales (beyond their correlation with one another) are not correlated with any other scales, apart from Dislike-Like which is negatively correlated with Interesting-Boring. This indicates that the two scales are in effect synonymous making one of them redundant. The lack of significant correlation with any of the other scales is again evidence of rating scales that are relatively ambiguous, This is perhaps due to the dependence on an individuals other interests in determining how boring a soundscape is. For example a committed ornithologist might find the recording at location 1, a forest environment with many birds present, to be very interesting in a way that someone apathetic to birds might not. In this way these scales could be seen as 'overly subjective' where it is not just where a stimulus rates on a scale that is a subjective point, but where the meaning of the scale itself also varies between individuals.

3.1.2 SAM Results

Regarding the other SD pairs used in the experiment, Figure 3 indicates that the scales Quiet-Noisy, Comforting-Discomforting, Pleasant-Unpleasant, Harmonious-Dissonant, Soft-Rough, Rural-Urban, and Calming-Agitating are all similar correlated with one another and can therefore be considered as representing the same rating scale (with Dislike-Like representing the same scale again but with reversed polarity).

Almost all of these scales are also negatively correlated with Valence (apart from Dislike-Like with which, as one might expect, it is positively correlated). This fact evidences that the Valence dimension of the SAM is just as informative as several of the SD pairs, meriting its future use as a replacement subjective measure.

Another positive result is the lack of significant corre-

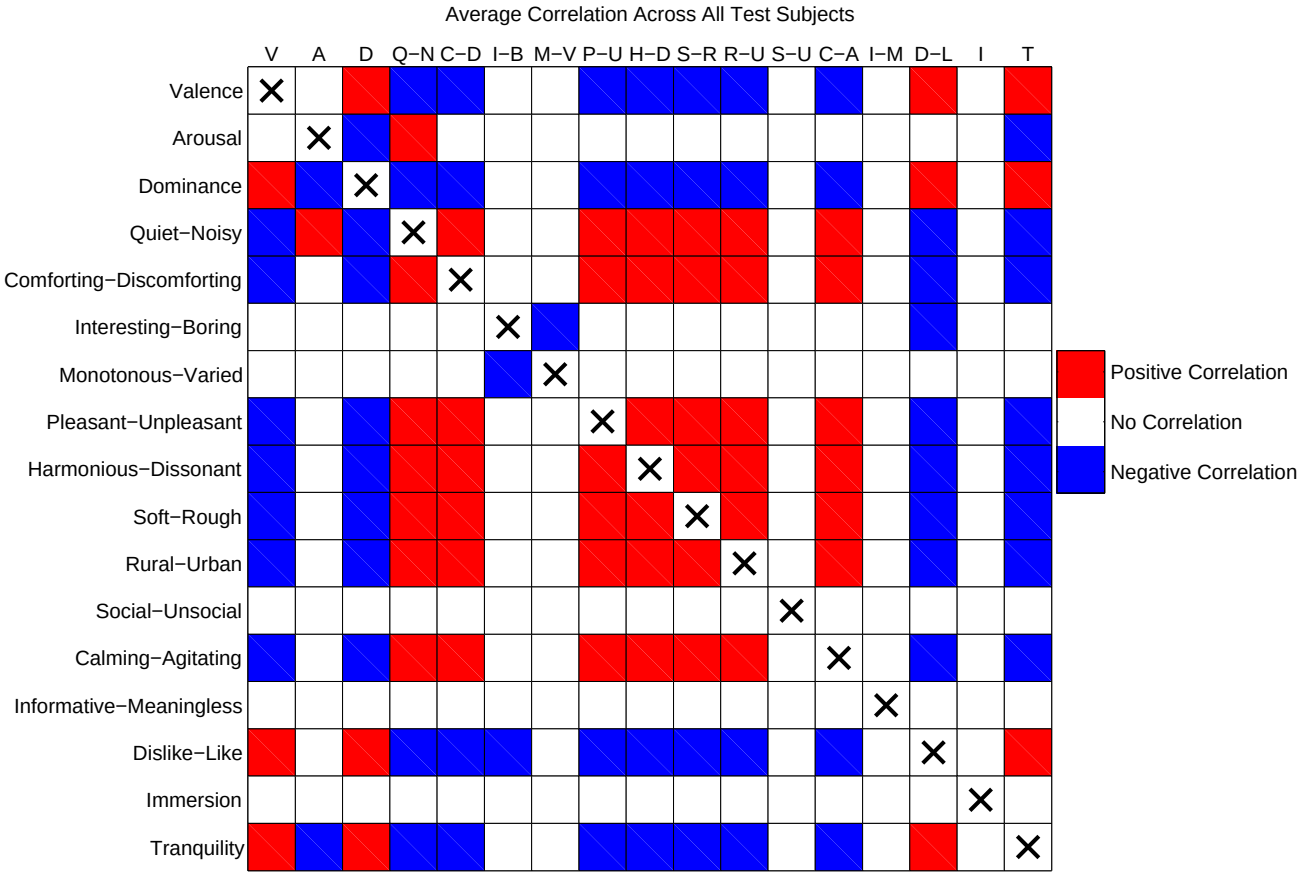


Figure 3. Plot showing the significant correlations between the rating scales used in the experiment. These values were calculated by taking an average of the correlation values between the rating scales for each participant, and then compared to the critical value for r described by the data.

lation between Valence and Arousal. The correlation of Arousal with the Quiet-Noisy and Tranquillity scales shows that the meaning of the Arousal scale has been correctly understood by the test participants, with the lack of significant correlation between Valence and Arousal indicating that the two scale are indeed measuring different elements of the subject experience, even if they can be indirectly related to one another due to their significant correlations with other rating scales. This justifies the future use of the Arousal dimension of the SAM in lieu of relevant SD pairs.

It is interesting to see in Figure 3 the significant correlation of Valence and Arousal with Dominance, as well as the correlation of Dominance with majority of the SD pairs. This is perhaps to be expected given Bradley and Lang's previous findings with the Dominance dimension of the SAM. They found that for the IAPS and IADS dimension that the meaning of the Dominance scale could be confusing; when rating the dominance of a photograph of, for example, a mutilated corpse the question arises as to whether the Dominance should be rated from the perspective of the viewer or the subject of the photograph [4].

In this case of this experiment then, the significant correlation of the Dominance with other ratings scales indicates that whilst it is a scale that might not provide too much information beyond that given by the Valence and Arousal scales, it is at least explicit to participants what the Dominance dimension means.

3.2 Circumplex Model of Affect

Another way of visualising the Valence and Arousal ratings for the soundscapes presented is to plot them as a Circumplex Model of Affect, a two-dimensional emotional space with arousal as one dimension and valence as another [5]. This is shown in Figure 4 where mean valence and arousal values for the set of clips have been (rescaled between ± 1) and plotted in 2D space.

The first thing apparent from Figure 4 is that the presented clips indicate a lack of spaces that are both valent and arousing. This leads to the question of whether there can be such a thing as a valent and arousing soundscape. This would require a busy soundscape with plenty of activity, but in a pleasant context (for the IAPS and IADS this has included examples such as erotica or a roller coaster). It remains to be seen, then, whether an example soundscape can be found that is both highy arousing and highly valent when presented in isolation, as the lack of visual context results essentially in increased aural arousal being equated with noise.

For these results then a pattern can be seen where increased arousal is associated primarily with the increased presence of traffic (as evidenced in Figure 4 by the differ-

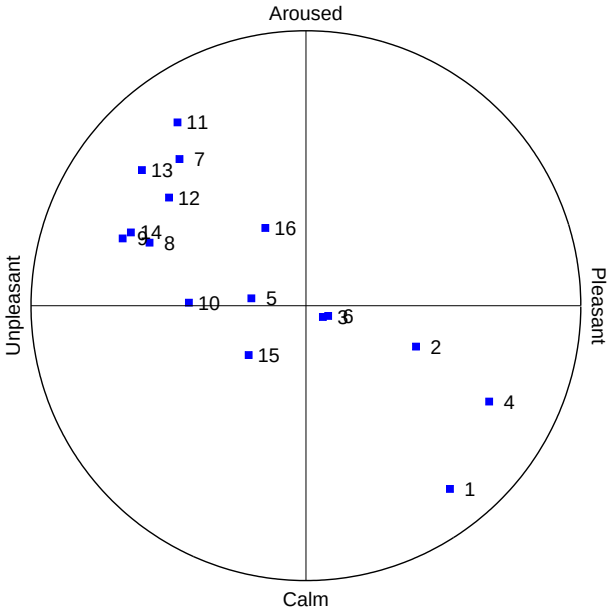


Figure 4. A plot of the mean arousal and valence values for each clip on the 'Circumplex Model of Affect', identifying their positions in 2D emotional space. The numbers correspond to the 16 recording clips used in the experiment, where there were two clips for each recording location. As such clips 1 and 2 represent location 1, clips 3 and 4 represents location 2, and so on up to location 8 (clips 15 and 16).

ence between the position of clip 3 and 4). Both of these clips were recorded at location 2 within 10 minutes of one another. The only difference between them is the presence of a single car driving by in clip 3 that is not present in clip 4. Future work using the visual data recorded at these locations will illuminate whether the pleasant visual setting will change subjective experience of an environment.

Fig. 4 also shows how the recording locations used can be separated into three broad categories:

- **Relaxing environments:** e.g. locations 1 and 2 in the bottom-right quadrant of the figure. These locations are situated in a rural forest environment, and contain the highest proportion of natural/animal sounds, as well as representing the lowest recorded SPL levels.

- **Neutrally rated environments:** such as locations 3 and 8 that inhabit the middle sections of the figure. These environments included a mixture of rural and urban features, and further experimentation will show how the visual features present in each environment may change positions in emotional space.

- **Stressful environments:** such as locations 6 and 7 that are placed in the upper-left quadrant. These recordings contain the highest proportion of traffic noise and other mechanical sounds, as well as representing the highest recorded SPL levels.

4. CONCLUSION

This paper has shown the results of an experiment comparing the use of SD pairs and the SAM for soundscape preference ratings. It was hypothesised that a comparison of soundscape preference rating results between a set of SD pairs and the SAM will show the SAM to be a directly comparable and equally useful tool for the analysis of subjective soundscape experience.

The correlation analysis results, as detailed in Section 3.1 and summarised in Fig. 3, have shown support for this hypothesis as the valence and arousal dimensions of the SAM correlate with the chosen SD pairs in such a way that they can be considered to explain the test subjects' responses to the data in a way that is just as meaningful as the SD pairs, but is less time consuming due to smaller number of rating scales. Dominance is shown to be or little use, which reflects findings from previous research and justifies abandoning its use in future experimentation. Feedback from test participants indicated that many felt the SAM was more intuitive to use than the SD pairs, which is borne out by the lack of significant correlation results for the Interesting-Boring, Social-Unsocial, and Informative-Meaningless rating scales.

The results shown in Section 3.2 indicate that the chosen recording locations do indeed cover a wide range of emotional ratings, but illuminate the lack of soundscape recordings that are both valent and arousing. It will be interesting to see whether the presentation of the recorded visual data alongside the soundscapes will change their perception and, accordingly, their positions in emotional space.

There are several further avenues of investigation to take in order to further explore the collected data. One is the analysis of biometric data collected alongside the subjective rating data examined in this paper, and the use of principal component analysis to further analyse the results presented here. The visual data recorded alongside the soundscape will also be presented to allow for the analysis of cross-modal perception. Another experiment has also been planned to compare SAM results with soundscape categorisation ratings assessing the recordings in terms of the three sound source groups identified in section 2.1.1.

Acknowledgments

This project has been supported by an EPSRC doctoral training studentship.

5. REFERENCES

[1] E. U., "Directive 2002/49/ec of the European Parliament and the council of 25 june 2002 relating to the assessment and management of environmental noise," *Official Journal of the European Communities*, vol. 189, no. 12, 2002.

[2] S. Harriet and D. Murphy, "Auralisation of an urban soundscape," *Acta Acustica united with Acustica*, vol. 101, no. 4, pp. 798–810, 2015.

[3] J. Kang and M. Zhang, "Semantic differential analysis

of the soundscape in urban open public spaces," *Building and environment*, vol. 45, no. 1, pp. 150–157, 2010.

[4] M. Bradley and P. Lang, "Measuring emotion: the self-assessment manikin and the semantic differential," *Journal of behavior therapy and experimental psychiatry*, vol. 25, no. 1, pp. 49–59, 1994.

[5] J. Russell, "A circumplex model of affect." *Journal of personality and social psychology*, vol. 39, no. 6, p. 1161, 1980.

[6] A. Léobon, "La qualification des ambiances sonores urbaines," *Natures-Sciences-Sociétés*, vol. 3, no. 1, pp. 26–41, 1995.

[7] S. Viollon, L. C., and C. Drake, "Influence of visual setting on sound ratings in an urban environment," *Applied Acoustics*, vol. 63, no. 5, pp. 493 – 511, 2002. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0003682X01000536

[8] S. Harriet, "Application of auralisation and soundscape methodologies to environmental noise," Ph.D. dissertation, University of York, 2013.

[9] W. Yang and J. Kang, "Acoustic comfort and psychological adaptation as a guide for soundscape design in urban open public spaces," in *Proceedings of the 17th International Congress on Acoustics (ICA)*, 2001.

[10] L. Anderson, B. Mulligan, L. Goodman, and H. Regen, "Effects of sounds on preferences for outdoor settings," *Environment and Bevior*, vol. 15, no. 5, pp. 539–566, 1983.

[11] G. Watts and R. Pheasant, "Tranquillity in the scottish highlands and dartmoor national park–the importance of soundscapes and emotional factors," *Applied Acoustics*, vol. 89, pp. 297–305, 2015.

[12] R. Pheasant, K. Horoshenkov, G. Watts, and B. Barrett, "The acoustic and visual factors influencing the construction of tranquil space in urban and rural environments tranquil spaces-quiet places?" *The Journal of the Acoustical Society of America*, vol. 123, no. 3, pp. 1446–1457, 2008.

[13] R. Pheasant, M. Fisher, G. Watts, D. Whitaker, and K. Horoshenkov, "The importance of auditory-visual interaction in the construction of 'tranquil space'," *Journal of Environmental Psychology*, vol. 30, no. 4, pp. 501–509, 2010.

[14] R. Gifford and C. Ng, "The relative contribution of visual and auditory cues to environmental perception," *Journal of Environmental Psychology*, vol. 2, no. 4, pp. 275–284, 1982.

[15] Ö. Axelsson, "How to measure soundscape quality," in *Euronoise2015, Maastricht*, 2015.

[16] O. Rummukainen, J. Radun, T. Virtanen, and V. Pulkki, "Categorization of natural dynamic audiovisual scenes," *PloS one*, vol. 9, no. 5, p. e95848, 2014.

[17] N. Bruce and W. Davies, "The effects of expectation on the perception of soundscapes," *Applied Acoustics*, vol. 85, pp. 1–11, 2014.

[18] M. Raimbault, "Qualitative judgements of urban soundscapes: Questionning questionnaires and semantic scales," *Acta acustica united with acustica*, vol. 92, no. 6, pp. 929–937, 2006.

[19] M. Southworth, "The sonic environment of cities," *Environment and Behavior*, vol. 1, no. 1, p. 49, Jun 01 1969, last updated - 2013-02-22.

[20] P. Fröhlich, S. Egger, R. Schatz, M. Mühlegger, K. Masuch, and B. Gardlo, "Qoe in 10 seconds: Are short video clip lengths sufficient for quality of experience assessment?" in *Quality of Multimedia Experience (QoMEX), 2012 Fourth International Workshop on*. IEEE, 2012, pp. 242–247.

[21] C. Guastavino, B. Katz, J. Polack, D. Levitin, and D. Dubois, "Ecological validity of soundscape reproduction," *Acta Acustica united with Acustica*, vol. 91, no. 2, pp. 333–341, 2005.

[22] C. Osgood, "The nature and measurement of meaning." *Psychological bulletin*, vol. 49, no. 3, p. 197, 1952.

[23] T. Hashimoto and S. Hatano, "Effects of factors other than sound to the perception of sound quality," *17th ICA Rome, CD-ROM*, 2001.

[24] W. Davies, N. Bruce, and J. Murphy, "Soundscape reproduction and synthesis," *Acta Acustica United with Acustica*, vol. 100, no. 2, pp. 285–292, 2014.

[25] S. Viollon and C. Lavandier, "Multidimensional assessment of the acoustic quality of urban environments," in *Conf. proceedings "Internoise", Nice, France, 27-30 Aug*, vol. 4, 2000, pp. 2279–2284.

[26] A. Zeitler and J. Hellbrück, "Semantic attributes of environmental sounds and their correlations with psychoacoustic magnitude," in *Proc. of the 17th International Congress on Acoustics [CDROM], Rome, Italy*, vol. 28, 2001.

[27] B. Schulte-Fortkamp, "The quality of acoustic environments and the meaning of soundscapes," in *Proc. of the 17th international conference on acoustics*, 2001.

[28] M. Bradley and P. J. Lang, *The International affective digitized sounds (IADS): stimuli, instruction manual and affective ratings*. NIMH Center for the Study of Emotion and Attention, 1999.

[29] M. Bradley and P. Lang, "Affective reactions to acoustic stimuli," *Psychophysiology*, vol. 37, no. 02, pp. 204–215, 2000.

[30] M. Bradley, B. Cuthbert, and P. Lang, "Picture media and emotion: Effects of a sustained affective context," *Psychophysiology*, vol. 33, no. 6, pp. 662–670, 1996.

[31] K. Hume and M. Ahtamad, "Physiological responses to and subjective estimates of soundscape elements," *Applied Acoustics*, vol. 74, no. 2, pp. 275–281, 2013.

[32] R. Hull and A. Harvey, "Explaining the emotion people experience in suburban parks," *Environment and behavior*, vol. 21, no. 3, pp. 323–345, 1989.

[33] K. Hanyu, "Visual properties and affective appraisals in residential areas in daylight," *Journal of Environmental Psychology*, vol. 20, no. 3, pp. 273–284, 2000.

[34] "Jury test analysis," 2015. [Online]. Available: http://www.salford.ac.uk/computing-science-engineering/research/acoustics/psychoacoustics/sound-quality-making-products-sound-better/sound-quality-testing/assessment-methods/jury-testing/10

[35] J. G. Peatman, *Introduction to applied statistics*. Harper & Row, 1963.

[36] "Calculate r critical," 2015. [Online]. Available: http://blog.excelmasterseries.com/2014/05/pearson-correlation-r-critical-and-p.html

[37] R. Schafer, *The Soundscape: Our Sonic Environment and the Tuning of the World*. Inner Traditions/Bear, 1993. [Online]. Available: http://books.google.co.uk/books?id=ltBrAwAAQBAJ

All online sources last accessed 24/06/16.

EMERGING COMPOSITION: BEING AND BECOMING
AN EXPERIMENT IN PROGRESS

Sever Tipei
Computer Music Project
University of Illinois
s-tipei@illinois.edu

ABSTRACT

Emerging Composition: Being and Becoming envisions a work in continuous transformation, never reaching an equilibrium, a complex dynamic system whose components permanently fluctuate and adjust to global changes. The process never produces a definitive version, but provides at any arbitrary point in time a plausible variant of the work - a transitory being. Directed Graphs are used to represent the structural levels of a composition (vertices) and the relationships between them (edges); parent-children and ancestor-descendant type connections describe well potential hierarchies in a piece of music. By determining adjacencies and degrees of vertices and introducing weights for edges, one can define affinities and dependencies in the complex and flexible structure that is a musical composition. Ways in which the all-incidence matrix of a graph with weighted edges can evolve are discussed including the use for that purpose of elements of Information Theory. The Emerging Composition model is closer to the way composers actually write music and refine their output; it also creates the equivalent of a live organism, growing, developing, and transforming itself over time.

1. BACKGROUND

The process of writing a new piece involves balancing elements that belong to different structural levels from the overall form of the composition to various sound characteristics. Composer Aurel Stroe and his collaborators discussed in the article "Morphogenetic Music" [1] the play between melody, rhythm, harmony, and phrase length in Mozart's *Piano Sonata in C Major K.W. 309* showing how unexpected or more daring choices at one structural level are compensated by blander, more familiar occurrences at others. A related insight into the composition process is given by Beethoven's sketchbooks that show a constant adjustment, sometimes over years, of initial motives [2] and by the works of Charles Ives who continued to modify his music even after it was published.

These universal concerns also apply to contemporary works and are shared by artists regardless of aesthetics, historical moment or style. Today, in electro-acoustic music, readily available software allows authors to easily investigate alternatives in placing and replacing gestures, textures, structural elements, etc. or even to further adapt and polish the sound materials after the completion of the project.

1.1 Manifold Compositions

When a computer-generated piece contains elements of indeterminacy, multiple variants can be produced simply by changing the initial conditions (eg. the random number generator's seed). Randomness may be involved in selecting the order of macro and micro events, in the choice of attack times and durations of sounds, of their frequencies, amplitudes, spectra, etc. or of their environment's properties such as location in space and reverberation. Such multiple variants, members of a *manifold composition*, have exactly the same structure and are the result of precisely the same process but differ in the way individual events with their diverse characteristics are distributed in time: like faces in a crowd, they all share basic features but exhibit particular attributes. A *manifold composition* is an equivalence class, a composition class, produced by a computer under particular conditions [3]. It includes all its actual and virtual variants and requires that all of them be equally acceptable. *Manifold compositions* build on the example of Stockhausen (*Plus-minus*) [4], Xenakis (*ST* pieces) [5] and Michael Gottfried Koenig (*Segmente*) [6] and extend it: by stipulating the use of a computer and introducing an element of indeterminacy during the act of composing in the case of Stockhausen; by adding more constraints in the last two cases.

1.2 DISSCO

The software used in the production of *manifolds*, a Digital Instrument for Sound Synthesis and Composition, DISSCO [7], provides a seamless approach to composition and sound design. An integrated environment, it has three major parts: LASS, a Library for Additive Sound Synthesis, which builds sounds from first principles (sine waves), CMOD, or Composition MODule, a collection of methods for composition that drives the synthesis engine, and LASSIE, a graphic user interface (GUI).

DISSCO is comprehensive in the sense that it does not require the intervention of the user once it begins to run. This kind of "black box" set of instructions is necessary for preserving the integrity of *manifold* production: intervening during computations or modifying the output would amount to the alteration of the data or of the logic

embedded in the software. Due to a LASS option not available on other systems, the precise control of the *perceived loudness* - a non-linear function of amplitude [8], post-production interventions become not only unnecessary but also incongruent with the purpose of the enterprise.

1.3 Indeterminacy

Randomness in DISSCO is introduced through simple uniform (flat) distributions by the RANDOM and RANDOMINT methods or made available through envelopes, functions of time (in most cases) built by the user. An envelope library, ENVLIB allows the composer to draw the contour of the curve, scale and store it while MAKE ENVELOPE offers not only the possibility to enter a list of ***x*** and ***y*** values but also to specify a range within which each of them may randomly fluctuate. Stochastic distributions expressions are handled through *muparser* [9], downloaded from the web and now part of DISSCO.

Two alternative options are introduced by STOCHOS: 1) a dynamic range whose min. and Max. limits are defined by two functions/envelopes while a third one controls the distribution of elements within the confined area and 2) multiple probability ranges whose sum is 1 at any moment (inspired by Xenakis' special density diagram determining orchestral composition)[10]. Finally, VALUEPICK, connected to SIEVE, introduces weighted probabilities assigned to discrete values at any parameter.

One of the underlying ideas behind CMOD is that when choosing concrete values from frequency and spectra to the formal design of a work, the same type of operations are used at different time scales. All these methods are considered "utilities" available in a multitude of situations.

2. DIRECTED GRAPHS

The structure of CMOD can be represented as a directed graph (DG), a *rooted tree,* where every structural level inherits from a generic Event class in a *matryoshka* type of arrangement: a unique Top event (the root) can include High events followed by Mid, Low, and Bottom events - the platform where individual sounds are created. In this model, events are represented as vertices each of them having siblings (except the root) and spawning any number of children. They are connected by edges that illustrate the relationships between them. By carefully determining adjacencies and degrees of all vertices and by introducing weights for edges, one can start defining affinities and dependencies in the complex and flexible structure that is a musical composition. The scheme can accommodate both the stricter order found in traditional music (piece < sections < themes < motives < cells < sounds), and, at the other extreme, if only the root and its immediate children are present, random distribution of undifferentiated events within the confines of the piece (sounds in Cage's chance music works). Moreover, this model is well suited to create "floating hierarchies", unstable flows of information that favor change over established formulations [11].

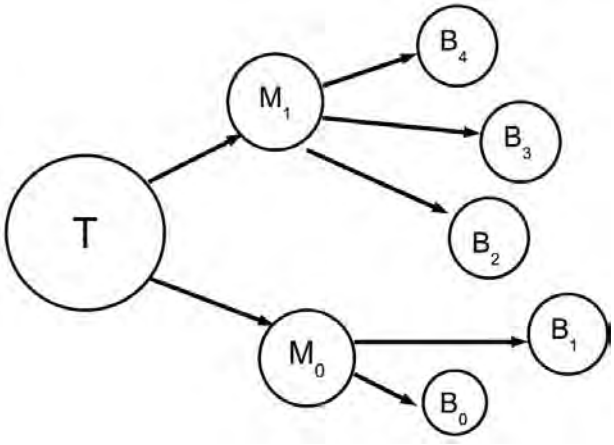


Figure 1. DISSCO structure as a rooted directed graph. For clarity only one intermediate level (M) is shown.

2.1 Similar Approaches

It should be noted that Pierre Barbaud had explored the use of graphs in "automatizing" the production of tonal harmonic and contrapuntal sequences in his own compositions as early as the 1960s [12] and that there is a similarity between DGs and the arborescences on which many later works of Xenakis are predicated. More recently, a number of authors have either proposed musical formalisms and/or built systems based on Directed Graphs. Among them, *Nodal* a system for generative composition [13] and *Graph Theory*, a piece by Jason Freeman [14] are the closer to the tenor of this project.

At the present time, Emerging Composition project adopts the point of view that one way in which any musical composition and its structural levels can be described is as a rooted tree DG. It is a framework that corresponds *post factum* to the way CMOD was organized, a starting point informed by musical practice. Not intended as a way to generate pitches, rhythms, etc. or to explore the limits of creativity like other schemes, it is used to represent relations between structural components of a musical work - a rather limited goal for this phase of the project that could be expanded in the future.

Relevant to possible future developments is Jonathan Owen Clark's formalism exposed in *Nonlinear Dynamic of Networks* [15] that brings together Graphs and Dynamic Systems. It could be applied to the content of vertices - sounds in a multidimensional vector space - and to their influence on the macro levels of a piece.

3. COMPLEX DYNAMIC SYSTEMS

Any composition can be thought of as a complex system. During the process of composing it, the system is also dynamic in the sense that options are re-evaluated at various times leading to changes both in the macro structure and in the details of the work. It should also be pointed out that, although complex dynamic systems are often assumed to be chaotic, that is not necessarily the case.

The Evolving Composition project models such a dynamic system by allowing the computations to continue for an arbitrary amount of time. It envisions a work in perpetual transformation, never reaching an equilibrium, a complex structure whose components permanently fluctuate and adjust to each other's modifications - a "brewing" piece. Such a composition can be regarded as a network of evolving interdependent elements whose alterations, refinements, and transformations create a series of unstable states. It could be likened to an electric grid where power is generated, routed, and distributed through different nodes: a network of diverse but interdependent components. The grid has to be responsive and to constantly adjust the flow of electricity to compensate for surges in demand or for local failures. Its musical equivalent is a composition whose parts are interconnected at all levels in such a way that modifying one component could have global consequences and affect other parts of the system.

This view of the composition as a network of perpetually unfolding elements in search of an elusive balance, similar to a living creature, epitomizes an "organic" approach to creating music. The process never produces a definitive version but provides at any arbitrary point in time a plausible variant of the work - a transitory being.

Emerging Composition: Being and Becoming is an augmentation and a corollary of the *manifold* idea as they both generate an unlimited number of variants, involve the presence of randomness at all structural levels, and relay on the view of sounds as events in a multidimensional vector space whose degrees of freedom include time/duration, frequency, amplitude, phase, etc. The project adopts the view that a composition could be represented as a hierarchical structure (but does not have to) and that it is predicated on discovering and creating new situations as opposed to attaining known, already established goals: a volatile, temporary equilibrium and NOT a search for a stable optimal solution.

4. THE DESIGN

4.1 Trivial Case

Upon finishing a new piece, a human composer might step back, take a fresh look at the work and, possibly, decide to make changes and adjustments. The Emerging Composition allows computations to continue after the first variant of the *manifold* is completed: a new edge is created between the last Bottom event $\mathbf{X}_{last}$, (a terminal vertex) and another vertex $\mathbf{X}_{new}$ which could be a sibling, a parent or an ancestor belonging to the same branch or to a different one. The operation takes place with the help of an all-incidence matrix $\mathcal{M}$ of the type shown in Figure 2.

This transitional matrix is weighted (expressing probabilities of exploring different edges) and serves as a template for the Evolving Entity, a sort of genome of the composition.

The selection of $\mathbf{X}_{new}$ involves dividing the components of the vector $\mathbf{V}_{last}$ (corresponding to $\mathbf{X}_{last}$) by their sum, adding the results in order from the top to bottom, with 1 in the last row, and matching a random number to one of the probability intervals thus created. If the newly chosen vertex $\mathbf{X}_{new}$ is a parent, all its descendents are computed anew. Upon completion an audio file becomes available for examination (or ignored) and a vector $\mathbf{V}_{new}$ corresponding to the chosen vertex $\mathbf{X}_{new}$ is used to continue. The procedure may be repeated an arbitrary number of times.

	T	**M_0**	**M_1**	**B_0**	**B_1**	**B_2**	**B_3**	**B_4**
T	0.01	0.05	0.05	0.02	0.01	0.03	0.03	0.01
M_0	0.20	0.01	0.25	0.20	0.20	0.05	0.10	0.07
M_1	0.20	0.01	0.01	0.11	0.10	0.25	0.20	0.23
B_0	0.12	0.20	0.11	0.01	0.30	0.19	0.08	0.07
B_1	0.12	0.24	0.10	0.35	0.01	0.07	0.08	0.09
B_2	0.12	0.06	0.15	0.10	0.12	0.01	0.26	0.26
B_3	0.12	0.08	0.18	0.11	0.14	0.25	0.01	0.26
B_4	0.11	0.06	0.15	0.11	0.12	0.25	0.24	0.01

Figure 2. All-incidence weighted matrix

4.2 Continuity

If the process of re-evaluating vertices proceeds without interruption, the continuous sequence of pseudorandom numbers creates a history uniquely determined by the seed state and its integrity confers to the Emerging Composition Entity in question the equivalent of a perennial "personality", an identity and a "individual history". There is a paradox here: the choices leading to any given variant of the manifold depend on chance but the random numbers themselves are part of a causal, deterministic chain. Combined with the fact that the directed graph and the matrix - the genome - are pre-determined, a balance is created between Being/structure, and Becoming/indeterminacy, and the piece starts to resemble a living organism whose cells are rejuvenated constantly while the creature endures.

4.3 Template Modification

Modifications of the template/genome may be introduced as the computations continue. If the column vector representing the last choice $\mathbf{V}_{last}$ is multiplied by the matrix $\mathcal{M}$, $\mathbf{V}_{last} * \mathcal{M}$, a Markov chain mechanism is initiated and the newly resulting vector $\mathbf{V}_{last+1}$ becomes part of an ordered sequence of causally connected vectors. This operation is repeated every time a new variant of the piece completes. In most cases the root of the tree, the piece itself, is not affected; however, that might change if the total duration of the entire piece is allowed to fluctuate between certain limits using one of the methods outlined in 1.3.

The user controls the likelihood of various connections/ edges between vertices through the static, all-incidence matrix $\mathcal{M}$. The Markov chain mechanism described above allows a vector to evolve in a predictable way but assumes that the content of the other vector/columns of the matrix remain the same. A more realistic alternative is to take into account global changes that might occur

every time a new version is computed - something a human composer would probably do.

Such adjustments are construed as the result of the composer's intuition, taste, training, etc. but many times these subjective considerations can also be described using elements of Information Theory. The main concepts provided by Information Theory as applied to musical messages are those of Entropy/Order - expressed through the relationship between Originality and Redundancy, a dialectical opposition - in relation to the Complexity of the work [8]. Their relevance to this project is based on at least two facts: these are measurable quantities and, as Herbert Brün once put it: "the job of a composer is to delay the decay of information".

As an example, Originality may be equated with i*m*probability hence with the delivered Information, Redundancy with repetition and/or familiarity, and Complexity with the number of available choices, all quantifiable if not entirely in an objective way. Since each variant of the piece exhibits new, different values for most vertices, an analysis of all values at all vertices followed by a comparison with a desired (dynamic) situation becomes necessary. In turn, such an extensive re-evaluation of data requires a significant increase in computing time and storage capacity since even a relatively short work may easily contain hundreds of vertices.

Moreover, the vertices representing the Bottom level contain significantly more information then those corresponding to higher level vertices and are more likely to trigger more often global changes. This is because sound design procedures are concentrated at the Bottom level: various ways of assigning the frequency and loudness of a sound, the rate and amplitude of vibrato (FM), of tremolo (AM) or of frequency and amplitude transients. Information about spatialization and reverberation should also be added to the list.

4.3 Developing Entity

The Complex Dynamic System that is the Emerging Entity/Composition includes the DG rooted tree that is DISSCO, the template/genome matrix $\mathcal{M}$, and the set of data used to create the initial variant of the piece. The preceding discussion has assumed the size of the rooted tree and, necessarily, that of the matrix, constant. However, the process could start with a tree and a matrix reduced to the smallest possible number of vertices/vectors, for instance only the Top vertex (the piece) and one or two Bottom or terminal vertices. The system is then allowed to grow, developing more edges and vertices at a rate controlled by the user until reaching its maximum potential. The opposite, a decaying slope can be engineered by cutting off branches of the tree and reducing gradually the size of the matrix. In the end, a restricted number of vertices and edges containing a smaller and smaller number of possible choices or a situation similar to reaching the ergodic (stationary) state of a Markov chain could signify the demise of the Entity/Composition. Using another analogy with biological processes, the growing number of vertices and edges in the beginning part of its evolution mirrored by a reduction of the network toward the end could be associated with the growing during the first years of human existence of the number of neurons and synapses and then the pruning that occurs during adolescence.

5. IMPLEMENTATION

5.1 Why DISSCO ?

Emerging Composition uses the structure and features of DISSCO, a powerful application that has been proven reliable and robust during an almost a decade of use. It also benefits from the experience accumulated both by seasoned users and by students in the classroom. DISSCO offers an unbroken link between a Computer-assisted (Algorithmic) Composition module that offers deterministic tools (patterns, sieves, etc.) along random distributions and a synthesis engine with uncommon capabilities (control of perceived loudness, polar coordinates, etc.) that generates - according to users - a sound output superior to many other similar applications. Combining complex designing abilities with a sophisticated artisanal proficiency and being an extension of the *manifold* undertaking, DISSCO constitutes the best choice available.

5.2 Present phase

After considering a number of alternatives, the general framework described above was selected. The Trivial Case was implemented by connecting the last Bottom event, a terminal vertex, to the Top event without interrupting the sequence of random numbers. This first stage of the project is used to generate *Sound Fountai*n, an installation generating continuous sound output in the atrium of a local modern building.

Presently, the Emerging Composition project runs in multithreading mode and has been recently ported on a multi-core system. Using 16 CPU cores when producing a complex eight channel piece, the ratio between computation time and the duration of the piece (real time) is a little less than 3/2 and increasing the number of cores does not result in a significant improvement. In other examples, a six minute stereo piece ran on the same system in less than five minutes while an experiment in granular synthesis, also done with DISSCO, of over 350,000 grains with dozens of partials each, took over five hours.

Computing time depends heavily on both the complexity of individual sounds and their individual duration. DISSCO was conceived as a "Rolls Royce bulldozer" (refined control over large numbers of elements) running on high-performance computers: it allows for an arbitrary number of partials and envelope segments along with involved selections of sound attributes. However, a meaningful functioning of the system requires a ratio of 1/1 or better and a urgent task is to profile, optimize, and parallelize the code in order to constantly achieve real time or faster.

5.3 Future work

As mentioned in the title, this is an experiment in progress and it is in its incipient stage. As previously described, a general plan has been formulated but many theoretical and practical aspects still need to be worked out.

Conceptually, the project is situated at the intersection of Dynamical Systems Theory, Graph Theory, and Information Theory and a link between them which is both solid and practical still needs to be formulated.

From a computational point of view, meaningful ways of creating the $\mathcal{M}$ matrix need to be explored. As an example, a recent work can be represented by a rooted tree containing 132 discrete vertices (event types). A 132 X 132 matrix or even matrices an order of magnitude higher are manageable but they will have to be constantly updated and various operations performed on them. In the case elements of Information Theory are used, an analysis of each new variant of the piece is necessary which means information for 15,700 sounds (as in the example mentioned) or more will have to be not only stored but also analyzed. Hence, the need for high-performance computing – at least at the present time.

6. CONCLUSIONS

Emerging Composition: Being and Becoming creates an original paradigm within the field of Computer-assisted (Algorithmic) Composition. Although the concepts of Complex Dynamic Systems and Graph Theory have been discussed in connection with Catastrophe Theory and Morphogenetic Music by Stroe [1] and by Clark [15] in the context of (pseudo-)tonal music [17], to our knowledge, there have been no proposals for building a concrete mechanism in order to generate a continuously evolving composition, resulting from uninterrupted computations.

This paradigm has the potential for further significant developments beyond the immediate scope of this project such as creating an "ecosystem" where the performance environment (hall acoustics) and live performers' decisions influence the composition - along the lines envisioned by Agostino Di Scipio [18].

6.1 Responding to well defined needs

The system described here responds to the needs of a personal creative project meant to expand the *manifold* undertaking. No other available software was deemed appropriate and, although its origins are idiosyncratic, it is maturing into a considerably more general tool. It creates pieces *in toto*, not piece-meal, preserving their identity in the case of *manifold* variants and insuring the continuity of the Evolving Entity through multiple generations. This is possible due to the unbroken flow between composition and synthesis merged into a unique smooth process that also insures total control of sound design details.

Similar to the *manifolds,* the system is based on the solid foundation provided by the description of sounds as events in a multidimensional vector space and it creates a general framework that can accommodate different aesthetics not just one particular style.

6.2 Pushing boundaries

Akin to some of the contributions already mentioned, this approach elevates the understanding of composition and composing to an abstract level and bridges the separation between music and other domains through the use of mathematical tools and by requiring state of the art, high-performance computing.

The Emerging Entity composition model is closer to how humans actually compose, by trial and error, continuously refining the output: it better approximates the workings of the human mind. It also reflects the natural world by creating (like some Artificial Life projects) the equivalent of a live organism, growing, developing, transforming itself over time and thus fulfilling the goal expressed by John Cage: " to imitate nature in its mode of operation".

In the process, it changes the role of the composer from artisan (making distinctive products in small quantities and using **traditional** methods) to that of a *Demiourgos* who creates from scratch (ie. using additive synthesis) multiple life-like Entities.

Emerging Composition and m*anifold compositions* represent an idiomatic way of using computers in music by mass spawning unique versions of the same archetype. A provision already present in the production of *manifolds* requires that a version of the output can not be performed in public more than once; presenting the piece as it exists only at one instance of a continuous process, an aspect of it which will be never repeated, stresses the ephemeral quality of any activity and prevents the piece to become a commodity. By questioning the uniqueness of the musical object - the piece - the proposal falls under the category of experimental music driven by speculative inclinations.

It also becomes the reflection of a particular worldview that contemplates the relationships between determinism/causality and randomness, between static templates and unforeseen particular events, between Being and Becoming. Or, in different words, it expresses the tension between the search for order and meaning and what Camus called the "silence of an indifferent universe".

Acknowledgments

We would like to acknowledge the support of Dr. Volodymyr Kindratenko and the Innovative Systems Laboratory at National Center for Supercomputing Applications (NCSA), for facilitating the computing infrastructure to perform the work.

7. REFERENCES

[1] A. Stroe, C.Georgescu, and M.Georgescu, ”Morphogenetic Music”, unpublished manuscript, Bucharest, cca. 1985.

[2] W. Kinderman, Artaria 195, Beethoven's Sketchbook for the Missa solemnis and the Piano Sonata in E Major, Opus 109, University of Illinois Press, Urbana, 2003.

[3] S. Tipei, “Manifold Compositions – a (Super)computer-assisted Composition Experiment in Progress”, Proc. 1989 Int'l Computer Music Conference, Ohio State University, Columbus, OH, 1989, pp. 324-327.

[4] K. Stockhausen, Plus-minus, Universal Edition, London, 1965

[5] I. Xenakis , ST pieces, Boosey and Hawkes, London, NewYork, 1967.

[6] M. G. Keonig, Segmente, Tonos Musikverlags GmbH, Darmstadt, Germany, 1983.

[7] H. G. Kaper and S. Tipei, “DISSCO: a Unified Approach to Sound Synthesis and Composition”, Proc. 2005 Int'l Computer Music Conference, Barcelona, Spain, September 2005, pp. 375-378.

[8] J. Guessford, H. G. Kaper, and S.Tipei. “Loudness Scaling in a Digital Synthesis Library”, Proc. 2004 Int'l Computer Music Conference, Miami, Florida, November 2004, pp. 398-401.

[9] - muparser - Fast Math Parser Library, http://beltoforion.de/article.php?a=muparser&p=features.

[10] I. Xenakis, Formalized Music, Pendragon Press, Stuyvesant, NY, 1992, p. 139.

[11] H. Brün, “On Floating Hierarchies”, talk given at American Society for Cybernetics, Evergreen College, October 20,1982,http://ada.evergreen.edu/~arunc/texts/brun/pdf/brunFH.pdf, accessed,September 7, 2015.

[12] P. Barbaud, Initiation a la composition musicale automatique, Dunod, Paris, 1966.

[13] J. McCormack, P. McIlwain, A. Lane and A. Dobrin, “Generative Composition with Nodal”, Workshop on Music and Artificial Life, Lisbon, Portugal, 2007.

[14] J. Freeman, Graph Theory, http://archive.turbulence.org/Works/graphtheory/, accessed April 23, 2016

[15] J. O. Clark, “Nonlinear Dynamics of Networks: Applications to Mathematical Music Theory”, in *Mathematics and Computation in Music*, Springer, Berlin, 2009, pp. 330-339.

[16] A. Moles, Information Theory and Aesthetic Perception, University of Illinois Press, Urbana, 1958.

[17] E. W. Large, “A Dynamical Systems Approach to Music Tonality”, www.ccs.fau.edu/~large/Publications/Large2010Tonality.pdf, accessed August 19, 2015.

[18] A. Di Scipio, “'Sound Is the Interface': from Interactive to Ecosystemic Sound Processing ”, Organized Sound, vol. 8(3), pp. 269-277, Apr. 2003.

OPTICAL OR INERTIAL? EVALUATION OF TWO MOTION CAPTURE SYSTEMS FOR STUDIES OF DANCING TO ELECTRONIC DANCE MUSIC

Ragnhild Torvanger Solberg
University of Agder, Department of Popular Music
ragnhild.t.solberg@uia.no

Alexander Refsum Jensenius
University of Oslo, Department of Musicology
a.r.jensenius@imv.uio.no

ABSTRACT

What type of motion capture system is best suited for studying dancing to electronic dance music? The paper discusses positive and negative sides of using camera-based and sensor-based motion tracking systems for group studies of dancers. This is exemplified through experiments with a Qualisys infrared motion capture system being used alongside a set of small inertial trackers from Axivity and regular video recordings. The conclusion is that it is possible to fine-tune an infrared tracking system to work satisfactory for group studies of complex body motion in a "club-like" environment. For ecological studies in a real club setting, however, inertial tracking is the most scalable and flexible solution.

1. INTRODUCTION

There has been a rapid growth in studies of music-related body motion over the last decades [1, 2], many of which have focused on musicians' sound-producing actions [3,4] or people's spontaneous motion to music [5,6]. Relatively few studies have been carried out on music–dance correspondences, and those have primarily focused on one or a few people dancing to music [7,8].

We are interested in studying (larger) groups of people moving to music, and to look more closely at the intersubjective relationships found in such music–dance settings. More specifically, we are looking at the relationship between body motions and the musical sound, and dancers' engagement with electronic dance music. Our long-term ambition is to carry out a large-scale experiment in a real club context. Due to ethical, practical and methodological challenges, however, we are currently running experiments in our controlled lab environment.

The aim of this paper is to present some of the challenges we have faced in setting up and running motion capture experiments with groups of 10–15 people dancing together. To our knowledge, few to none empirical studies have investigated how groups of people dance and relate to electronic dance music–even though this is a common and wide-spread form of engaging with this musical style. Therefore, we have realized the need to develop an ecologically valid motion capture research design that can be used to study such musical group behaviour in both a lab context and real-life settings.

We will start by presenting an overview of some relevant motion capture technologies. This is followed by brief presentations of the experiments we have conducted thus far and a discussion of solutions found to different technical and methodological challenges.

2. MOTION TRACKING TECHNOLOGIES

When it comes to systems for tracking human body motion, we may, very roughly, differentiate between two main types of technologies: camera-based and sensor-based systems. Each of these can further be subdivided into a number of categories. For the discussion here, however, we will consider three concrete solutions that in different ways could be used to capture dance motion:

- Video recordings, using a single, off-the-shelf video camera to record the entire dance space, followed by the application of computer vision techniques to extract relevant features
- Optical, infrared motion capture, using a setup with multiple infrared cameras to record the position of reflective markers on the body of the dancers
- Small inertial sensor devices with built-in accelerometers, gyroscopes and magnetometers, recording directly to an on-device memory storage

In the following we will briefly discuss benefits and possible challenges of each type of system.

2.1 Regular video recordings

There are both theoretical and practical limitations when it comes to using regular video recordings as the basis for tracking human motion. Even though there has been enormous progress in the field of *computer vision* in recent years [9–11], this method is still limited to primarily tracking motion in two dimensions. That means that the position of the camera is crucial for the final result, since only what can be seen, can be tracked. If one wants to track the position of a group of dancing people, the most sensible camera position would be in the ceiling. Such a position allows for

capturing people's horizontal motion, but not much of the vertical displacement.

More specialized cameras, such as time-of-flight and stereo cameras, may allow for recording "pseudo-3D," or at least get some depth information. But it is only with multiple cameras surrounding the capture space that it is possible to carry out true 3D motion tracking from cameras.

Another problem with using video recordings for our type of studies is the influence of changing lights. An important element of a club environment is that of light and laser effects, rapidly moving lights with changing colours. This problem can be overcome to some extent by using cameras with some kind of "night mode," using a filter that only passes through the infrared light. The latter might entail the use of infrared light sources to function properly, which further complicates the setup.

The perhaps biggest challenge with regular video cameras, however, is that of actually tracking people or objects. Tracking an individual person can be hard enough if the background in the image is too noisy. Needless to say, it is quite a challenge to track individual people within a group of dancers in a dark setting and with changing lights. With this in mind, we never really tried to use regular video tracking for our current experiments, but rather used video recordings only for documentation and reference purposes.

2.2 Optical, infrared motion capture

While progress is being made for carrying out markerless, camera-based motion capture [12], the current state-of-the-art is still setups of multiple infrared cameras placed around the capture space, and with subjects wearing reflective markers. Using markers on the body allows for high spatiotemporal accuracy and precision of the joints being tracked. The use of infrared cameras with built-in light sources makes it possible to also track people in rooms with no, limited or changing light. One problem remains, though, the need for line-of-sight from cameras to markers.

Another challenge with infrared motion capture systems, is that they work best in a controlled laboratory setting. We have experienced numerous challenges when setting up the system outside of the lab, such as in a regular concert venue. First, carrying out the calibration process—moving a wand with reflective markers around the space—may be problematic if there are people present. This means that it would be necessary to calibrate the system before people arrive to the venue, which could possibly be several hours before an actual recording would take place.

A second challenge is that the calibration of the system could easily be ruined if any of the cameras move after the calibration has been performed. In our experience, it is a high risk for someone to bump into a camera stand or cable in a public space, which would result in the need for a new calibration to be performed. Even if we were to mount the cameras in the ceiling, the vibrations alone in a club space with loud music might very well be sufficient to require a re-calibration of the system.

Finally, infrared systems are very sensitive to reflections, everything from reflective materials on people's clothes, to bottles and glasses. Such reflections would end up as tracked markers in the system, thus complicating the tracking of individuals. In the best of cases, many such "ghost" markers would require a very long post-processing process to identify individual markers. In a worst case scenario, too many reflections could possibly ruin an entire data set.

To conclude, it may very well be theoretically possible to use an infrared system in a real club context, but due to the many practical challenges we have for now decided to work in our controlled lab environment.

2.3 Inertial sensor-based systems

Many of the challenges presented above are non-existent for systems based on *inertial* sensors. The two main types of inertial sensors are *accelerometers* and *gyroscopes*, and both of these sensor types are based on measuring the displacement of a small "proof-mass." Accelerometers measure the positional displacement of such a mass, while gyroscopes measure the rotational. By combining three accelerometers and three gyroscopes it is possible to capture both three-dimensional position and three-dimensional rotation in one small sensor unit.

One of the most compelling features of inertial sensors, is that they rely on physical laws (gravity), which are not affected by external factors, such as lighting. They can also be made into very small and self-contained units, with low power consumption and high sampling rates. These are probably some of the reasons why inertial sensors are now becoming integrated in a lot of technologies, further propelling down the cost of single units and securing even broader integration in all sorts of electronic devices.

The downside to inertial sensing is that accelerometers do not measure the *position*, but rather the *rate of change* of the subjects. It is possible to estimate the position through integration, and, combined with the data from gyroscopes and magnetometers, this can lead to satisfactory results [13]. However, while the relative position estimates may be good, such position data often suffer from a considerable amount of drift [14]. One way to overcome some of the drift problems in inertial systems, is by adding other sensor types and possibly also cameras [15]. This is common in more advanced inertial motion capture systems, but is not possible with smaller and cheaper integrated units.

3. DANCE EXPERIMENTS

Due to the many challenges of working in a real-life setting, we decided to carry out our current studies in a (motion capture) lab environment. Still we wanted to make the experiments as ecological as possible, so care was taken in transforming the lab into a "club-like" environment. The club setting is characterized by many people dancing relatively close to loud music in a darkened space with light effects. Therefore, we covered all the lab's walls in black, turned off the lights, and added various changing light effects. The 60-channel sound system secured an immerse sound experience. Thus the final visual and audible appearance was comparable to that of a club setting (Figure 1).

We will in the following briefly describe the two experiments we have conducted so far. Our focus is on method-

Figure 1. The fourMs motion capture lab at the University of Oslo: 1) before light adjustments, 2) after light adjustments, and 3) during the dance session.

ological considerations, as the results of the experiments will be published elsewhere [16, 17].

3.1 Dance Experiment 1

The first experiment was carried out in June 2014, with 16 people participating in a 15-minute long dance session. In this experiment we used a high-quality infrared, marker-based system from Qualisys, with nine Oqus 300 cameras [1] surrounding the capture space and running at 100 Hz. The system was for this experiment calibrated at the level of the floor. Each subject was equipped with two reflective markers: one positioned on the head and another on one of the wrists. The initial idea was to capture both general motion patterns (from the head) and more local activity (from the arm) of the subjects while dancing.

Even though we had done several smaller pilot studies prior to the actual experiment, we ended up with a lot of tracking problems. The biggest challenge was the large dropout rate of the wrist markers, since the subjects danced so close to each other that the markers were covered up most of the time. We also experienced challenges with reliably tracking the head markers due to people raising their arms and shifting positions while dancing. The raised arms covered many of the head markers, and when the wrist markers came close to the head markers, it also confused the proximity-based trajectory detection. Other markers disappeared for some time when some dancers bent down and danced close to the floor, and others when they moved around in the space.

All in all, the tracking percentage of the head markers was on average quite good, even though there were too many broken trajectories to reliably track individual subjects throughout an entire recording. Thus the final data set could not be used for the individual analysis that we had originally hoped for, but it still presented a solid and useful data set with the possibility to estimate the general "quantity of motion" of all the subjects.

3.2 Dance Experiment 2

The knowledge gained from the first experiment was vital when planning our second experiment, which was carried out in August 2015. Here we did several adjustments and updates to the research design. First, we decided to put a limit on 10 participants at a time, so the 29 recruited participants were distributed in three groups. Even though this is a somewhat smaller group than in the first experiment, it can still be qualified as a sufficient amount of people to simulate an ecologically valid dance setting. Also, by using smaller, but several groups, we had the added benefit of looking at differences and similarities between groups.

Figure 2. Calibrating the 20 AX3 sensors, by moving all of them rapidly up and down in a synchronization routine.

To ensure the recording of at least one good data set, we decided to do parallel recording with all the three techniques mentioned in Section 2: infrared optical tracking, inertial sensors and regular video recording.

Several measures were taken to improve the quality of the infrared tracking. First, each subject was equipped with only one reflective marker, positioned on the head, to reduce the problem of marker occlusion and confusion. Nine Qualisys Oqus 300 cameras were hanging around the walls of the room, in the same configuration as for the first experiment. This time four additional cameras (Oqus 400) were positioned in the ceiling above the capture space. These additional cameras greatly improved the tracking percentage to nearly 100 % for each tracked subject. To reduce any possible measurement errors, the system was calibrated at head's height, approximately 1.6 m above the floor. We also increased the frame rate of the Qualisys system to 200 Hz, since it has recently been shown that the minimum frame rate needed to capture motion should not be chosen based on the Nyquist-Shannon theorem, but rather according to the ratio between the maximum speed and the minimum spacing between markers [18].

Since it had proven difficult to capture arm motion using the infrared system, we opted for using inertial sensing of the activity of the arms. Here the AX3 armband sensor unit from Axivity [2] was chosen. These sensors are made for long-term motion recording (up to one month continuously) and are running as standalone and individual modules. They each have an internal clock that is updated when connecting to a computer and this clock is used to record time-stamps to the data file. From our initial testing, we found that the clocks' time-stamps deviated too much to be used for synchronization. So the solution was to do a manual synchronization routine with a set of repeated non-periodic spikes with all devices at the same time after starting the devices and before stopping them (Figure 2).

[1] http://www.qualisys.com/cameras/oqus/

[2] http://www.axivity.com/product/ax3

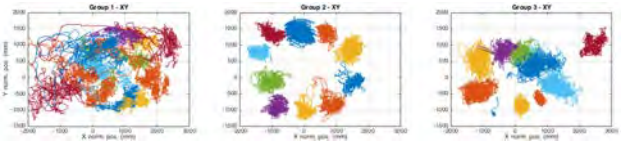


Figure 3. XY-plots from the infrared motion capture data, showing the horizontal motion patterns of subjects in each of the three groups.

To further fail-proof the setup, the sessions were recorded using four regular Canon prosumer cameras. They were placed in each of the labs corners so as to cover the scene from different angles. The cameras' "night-mode" setting was used to remove the changing light effects. The four video streams were run into a quad picture-in-picture video mixer, so that we could record a combined full HD video stream in the QTM software. This stream would carry audio as well, ensuring synchronization between audio, video and motion capture data.

It turned out that all of the three recording types worked well. The visual quality of the video recordings was quite poor, as expected when using the "night-mode" setting, but they were still useful for visual inspection and also worked well for some basic quantitative video analysis techniques. The data files from the Axivity sensors were flawless as soon as we managed to time-align them properly (see Section 4.3). We were also satisfied to see that all the efforts made in improving the setup of the infrared system paid off; the tracking of head markers from the Qualisys system was near-perfect, with a 100% fill level for most of the subjects. This made it, among others, possible to follow individual trajectories over time (Figure 3).

4. DISCUSSION

We will in the following discuss some technical issues relating to the quality of the recorded data, as well as reflect on the different systems' usage and possibilities.

4.1 Data Quality

While we ended up with a lot of broken motion trajectories in the data recorded with the Qualisys system in Experiment 1, the recordings from Experiment 2 resulted in near-perfect tracking results. This was the case even though people danced closely, moved around the space, jumped up and down, and held their arms in the air. The main reasons for the improved tracking results were probably a combination of having four cameras pointing down from the ceiling, the reduction of markers, fewer participants per group, and the increase in capture rate. Additionally, we believe that calibrating at 1.6 m above the floor level also helped to reduce possible measurement errors.

As expected, there were no problems with the data from the inertial sensors. An added benefit of using inertial sensors is that each device has a unique ID, which makes it possible to track individuals over time, even the ones that move around a lot in the space. We could also pair the inertial sensors to each of the subject's infrared marker. The downside to inertial systems, however, is that the devices measure relative motion (based on the gravitational pull) and not the exact location in space.

4.2 Spatiotemporal Resolution

It has been shown that the spatiotemporal resolution of a Qualisys system is much higher than what is needed for studying human body motion [19]. This is the case, even though the spatial accuracy and precision is uneven throughout the space [20]. It was the new knowledge about the proportionality between the speed and spacing of markers [18] that made us increase the frame rate from Experiment 1 to Experiment 2 (from 100 Hz to 200 Hz). So while such a frame rate is not necessary to capture the motion observed in the dancers, it clearly reduced the number of marker dropouts in the recordings.

It has been shown that the AX3 does not provide the same spatiotemporal accuracy and precision as the Qualisys system [21]. Still, the spatial resolution and data rate is more than sufficient for capturing the large-scale body motion seen in dance studies. An added benefit of inertial sensors is that they provide an even spatial accuracy and precision all over the recording space, as opposed to the infrared markers.

4.3 Synchronization to Audio

One of the most challenging parts when it comes to working with motion capture systems in a musical context, is the need for synchronizing motion data to related audio and video files. One of the positive sides of using a complete motion capture solution like that provided by Qualisys, is that it allows for SMPTE-based synchronization of cameras to an audio interface. This makes it possible to record high-quality audio and video with frame-based synchronization to the motion capture data.

The AX3 sensors, on the other hand, are standalone devices with no proper synchronization mechanism. The easiest solution is to use the built-in clocks for synchronization, but our tests have shown that they drift apart for longer recordings. This is negligible in many cases, but they are not accurate enough when we want to synchronize several hour-long recordings to rapid, beat-based music. Fortunately, the AX3s sample evenly, and can hence be synchronized based on reference points at the beginning and end of recordings.

Our solution was to carry out a manual synchronization routine (Figure 2) consisting of five non-periodic spikes created over a period of about 30–40 seconds. This synchronization routine was performed at the beginning and end of the experiment, with several hours in between. A simple cross-correlation algorithm in Matlab aligned the data sets based on the spikes at the beginning and end of each of the 20 data files. As the plots in Figure 4 show, the result was near-perfect time alignment of all the AX3 data. Since the routine was carried out in front of the video cameras, which also recorded audio, it was possible to use the spikes to time-align the AX3 data sets, which could then also easily be synchronized with the audio and video files.

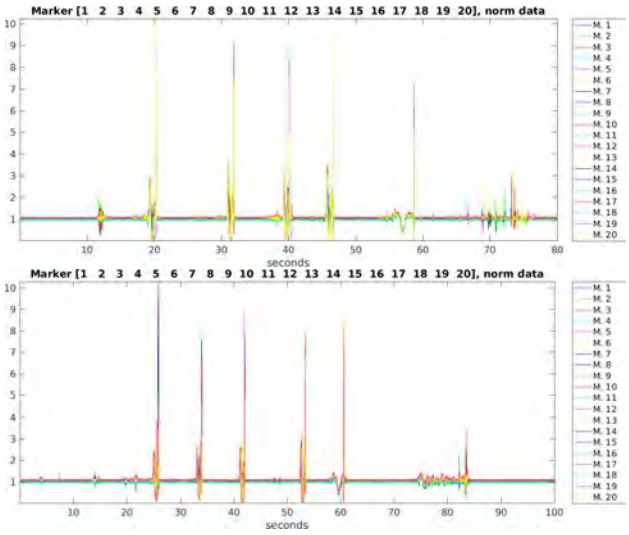


Figure 4. Plots of the synchronized calibration spikes at the beginning (top) and end (bottom) of a 2-hour recording.

4.4 The researcher's perspective

By carrying out these two dance experiments—in addition to several pilots in-between—we have gained valuable insights into how to study musical group behaviour with different types of motion capture systems. First, we found that a time-effective research setup is an important aspect when carrying out such group studies, particularly if one wants to get a group of 10 participants prepared for recordings in just a few minutes' time. This is especially important when thinking about conducting a large-scale study in a real club environment. We therefore spent time on testing how few markers we could use, and where to put them to not compromise the accuracy or quality of the data. Having a reduced set of sensors/markers that are easy to distribute and put on for the subjects themselves, greatly assist in the preparation time for an experiment.

Synchronization turned out to be a main concern in the design and performance of the experiments; both that the markers/sensors were in synchronization with each other and with related audio and video. Here we were particularly concerned with making a synchronization routine that could be carried out before and after a series of experiments, so as to not have to do any synchronization during the course of the experiments.

We were also satisfied to find that it was possible to carry out several hour-long recording sessions with the inertial sensors, with many sensors and subjects. This will be of importance for an actual club setting, during which recordings would typically go on for hours with many people present. Such long recordings would certainly not be possible with an infrared system, as the number of broken trajectories would be too large to handle.

A further important premise was the ecological validity, and that the systems in use should not attract too much attention or be too intrusive regarding the personal space of the subjects. We found that the ways we ended up applying the infrared markers, the inertial sensors and the video recording satisfied this specific premise.

4.5 The subject's perspective

Most people would find dancing together with others in a lab environment somewhat unnatural and awkward–at least in the beginning. The "clubification" of the lab certainly helped in creating a relaxed and natural atmosphere for the subjects. We received a lot of positive feedback from the people participating in the experiments about the layout of the lab. It also helped having sofas, music, food and non-alcoholic drinks outside the lab space, thus creating a social atmosphere surrounding the experiments.

The use of a very limited sensor/marker setup also helped in reducing the feeling of being part of an experiment. The reflective marker on the head is lightweight and barely noticeable when put on, and the AX3 sensor feels like a regular watch. It probably also helped that the participants could easily put the equipment on themselves. None of the participants commented that the sensors had invaded their personal space, and they quickly forgot about them as soon as they had put them on.

This shows that even though the participants are not dancing in an actual club space, it is, indeed, possible to carry out such group studies with a certain level of ecological validity in a mocap lab.

5. CONCLUSION

As far as we know, there have been few experiments using advanced motion capture systems with such a number of people and in such a noisy environment that we have attempted. After having experienced several tracking problems in Experiment 1, we obtained near-perfect infrared motion tracking results for all three groups dancing together in Experiment 2. We also managed to successfully beat-synchronize hour-long recordings of 20 inertial trackers through a simple calibration routine.

As expected, regular video recordings do not work very well for the experiments in question. Video recordings, even in full HD quality, have limited spatiotemporal resolution, and it is difficult to adequately track individuals in a larger group of people. That said, video recordings with "night-mode" turned on, are of high value for documentation purposes and for assisting in the post-processing of sensor and marker data of a large group of people.

Based on the knowledge gained from these experiments, we are currently planning new lab-based recordings using the infrared motion capture system to further investigate musical group behaviour. We are also one step closer to realizing a larger study with inertial sensors in a real club context.

Acknowledgments

Thanks to all the dancers that participated in the study, to Diana Kayser, Minho Song and Mari Romarheim Haugen for their assistance in running the experiments, and to Kristian Nymoen for help with creating the AX3 synchronization routine in Matlab.

6. REFERENCES

[1] M. M. Wanderley and M. Battier, Eds., *Trends in Gestural Control of Music*. Paris: IRCAM - Centre Pompidou, 2000. [Online]. Available: http://www.music.mcgill.ca/mwanderley/Trends/

[2] R. I. Godoy and M. Leman, Eds., *Musical Gestures: Sound, Movement, and Meaning*. New York: Routledge, 2010.

[3] A. Gritten and E. King, Eds., *Music and gesture*. Hampshire: Ashgate, 2006.

[4] ——, *New Perspectives on Music and Gesture*. Hampshire: Ashgate, 2011.

[5] K. Nymoen, B. Caramiaux, M. Kozak, and J. Torresen, "Analyzing sound tracings: a multimodal approach to music information retrieval," in *Proceedings of the 1st international ACM workshop on Music information retrieval with user-centered and multimodal strategies*. New York, NY: ACM, 2011, pp. 39–44. [Online]. Available: http://dx.doi.org/10.1145/2072529.2072541

[6] B. Burger, "Move the way you feel: effects of musical features, perceived emotions, and personality on music-induced movement," PhD thesis, University of Jyvskyl, 2013. [Online]. Available: https://jyx.jyu.fi/dspace/handle/123456789/42506

[7] L. Naveda, "Gesture in Samba: A cross-modal analysis of dance and music from the Afro-Brazilian culture," Ph.D. dissertation, Ghent University, Jan. 2011. [Online]. Available: http://www.ipem.ugent.be/samba/SambaProject/Thesis_files/Naveda2011_GestureInSamba_PhDthesis.pdf

[8] M. R. Haugen, "Studying Rhythmical Structures in Norwegian Folk Music and Dance Using Motion Capture Technology: A Case Study of Norwegian Telespringar," *Musikk og Tradisjon*, vol. 28, 2014. [Online]. Available: http://ojs.novus.no/index.php/MOT/article/view/688

[9] T. B. Moeslund and E. Granum, "A Survey of Computer Vision-Based Human Motion Capture," *Computer Vision and Image Understanding*, vol. 81, no. 3, pp. 231–268, 2001. [Online]. Available: http://linkinghub.elsevier.com/retrieve/pii/S107731420090897X

[10] T. B. Moeslund, A. Hilton, and V. Krger, "A survey of advances in vision-based human motion capture and analysis," *Computer Vision and Image Understanding*, vol. 104, no. 2-3, pp. 90–126, 2006. [Online]. Available: http://linkinghub.elsevier.com/retrieve/pii/S1077314206001263

[11] S. S. Rautaray and A. Agrawal, "Vision based hand gesture recognition for human computer interaction: a survey," *Artificial Intelligence Review*, vol. 43, no. 1, pp. 1–54, 2015. [Online]. Available: http://link.springer.com/article/10.1007/s10462-012-9356-9

[12] L. Sigal, A. Balan, and M. Black, "Humaneva: Synchronized video and motion capture dataset and baseline algorithm for evaluation of articulated human motion," *International journal of computer vision*, vol. 87, no. 1, pp. 4–27, 2010. [Online]. Available: http:/dx.doi.org/10.1007/s11263-009-0273-6

[13] H. Wilmers, "Bowsense an Open Wireless Motion Sensing Platform," in *Proceedings of the International Computer Music Conference*, Montreal, Canada, 2009, pp. 287–290. [Online]. Available: http://hdl.handle.net/2027/spo.bbp2372.2009.064

[14] S. Skogstad, K. Nymoen, and M. E. Hvin, "Comparing Inertial and Optical MoCap Technologies for Synthesis Control," in *Proceedings of Sound and Music Computing*, Padova, Italy, 2011, pp. 421–426. [Online]. Available: http://smcnetwork.org/system/files/smc2011_submission_124.pdf

[15] G. Welch and E. Foxlin, "Motion tracking: no silver bullet, but a respectable arsenal," *IEEE Computer Graphics and Applications*, vol. 22, no. 6, pp. 24–38, 2002. [Online]. Available: http://dx.doi.org/10.1109/MCG.2002.1046626

[16] R. T. Solberg and A. R. Jensenius, "Pleasurable and Intersubjectively Embodied Experiences of Electronic Dance Music," *Empirical Musicology Review*, 2016.

[17] ——, "Group Behaviour and Interpersonal Synchronization to Musical Passages of Electronic Dance Music."

[18] M.-H. Song and R. I. Gody, "How Fast Is Your Body Motion? Determining a Sufficient Frame Rate for an Optical Motion Tracking System Using Passive Markers," *PLOS ONE*, vol. 11, no. 3, p. e0150993, Mar. 2016. [Online]. Available: http://journals.plos.org/plosone/article?id=10.1371/journal.pone.0150993

[19] A. R. Jensenius, K. Nymoen, S. Skogstad, and A. Voldsund, "A Study of the Noise-Level in Two Infrared Marker-Based Motion Capture Systems," in *Proceedings of the Sound and Music Computing Conference*, Copenhagen, 2012, pp. 258–263. [Online]. Available: http://www.duo.uio.no/sok/work.html?WORKID=167469&fid=100544

[20] G. Vigliensoni and M. M. Wanderley, "A Quantitative Comparison of Position Trackers for the Development of a Touch-less Musical Interface," in *Proceedings of the International Conference on New Interfaces for Musical Expression*, Ann Arbor, MI, 2012. [Online]. Available: http://www.nime.org/proceedings/2012/nime2012_155.pdf

[21] M. R. Haugen and K. Nymoen, "Evaluating Input Devices for Dance Research," in *Proceedings of the International Symposium on CMMR*, Plymouth, UK, 2015, 00000. [Online]. Available: http://www.researchgate.net/profile/Mari_Haugen/publication/280830590_Evaluating_Input_Devices_for_Dance_Research/links/55c867e408aeb97567470848.pdf

CAMEL: CARNATIC PERCUSSION MUSIC GENERATION USING N-GRAM MODELS

Konstantinos Trochidis
New York University
Abu Dhabi
kt70@nyu.edu

Carlos Guedes
New York University
Abu Dhabi
carlos.guedes@nyu.edu

Akshay Anantapadmanabhan
Independent
Musician
akshaylaya@gmail.com

Andrija Klaric
New York University
Abu Dhabi
ak4867@nyu.edu

ABSTRACT

In this paper we explore a method for automatically generating Carnatic style rhythmic. The method uses a set of annotated Carnatic percussion performances to generate new rhythmic patterns. The excerpts are short percussion solo performances in ādi tāla (8 beat-cycle), performed in three different tempi (slow/moderate/fast). All excerpts were manually annotated with beats, downbeats and strokes in three different registers — Lo-Mid-Hi. N-gram analysis and Markov chains are used to model the rhythmic structure of the music and determine the progression of the generated rhythmic patterns. The generated compositions are evaluated by a Carnatic music percussionist through a questionnaire and the overall evaluation process is discussed. Results show that the system can successfully compose Carnatic style rhythmic performances and generate new patterns based on the original compositions.

1. INTRODUCTION

Automatic generation of music has been a focus of computational music research for a long time. Researchers have been designing systems to imitate or compose various musical styles from Classical to Jazz music [1], [2]. Despite the progress achieved so far in the development of generative music systems for Western music genres there is limited work regarding methodologies of automatic generation of music in non-western styles.

In this paper, we propose CAMeL an automatic music generation system, which focuses on the generation of Carnatic style rhythms. Carnatic music is an art music tradition from South India with a long history, which has its own musical grammar and significant musicological literature [3]. Carnatic music has a very well defined rhythmic framework and an interesting rhythmic structure, which makes it interesting and challenging to explore in an automatic music generation system. The approach proposed in this paper is focused on percussion-based Carnatic music style rhythms using a set of annotated training data of music excerpts. The annotations include the stroke register (Lo-Mid-Hi), the inter-onset interval duration of the strokes and the amplitude of the music excerpts. By extracting these features the system is capable of automatically generating new rhythmic progressions stylistically similar to the training compositions. N-gram analysis and statistical learning is used to model the rhythmic structure using the extracted features. Markov chains are then used to build the rhythmic development and describe the pattern transition likelihoods of the generation sequences. The system generates rhythmic patterns based on an n-gram input. If a five-gram analysis is selected then the algorithm generates the strokes using the transition probability of the five-grams.

The proposed method for generating rhythmic pattern progression of Carnatic style music was evaluated by a professional Carnatic percussionist — Akshai Anantapadmanabhan. The same percussionist composed and performed the datasets for training the system. The evaluation is based on feedback of the rhythmic structure and development of the generated sequences compared to a human-based performance. The results of the evaluation provide insights into the rhythmic organization and interpretation of the generated rhythmic patterns.

Musicians can use the proposed system for creative purposes in their performance and training. It can be also used as a tool in music education as a means of actively enculturing lay people into this music style; for example, by creating software applications that include generative systems of Carnatic music, allowing users to “play” Carnatic music percussion on mobile devices and get entrained in this style by getting familiar with the underlying rhythmic structure and grammar of this music.

The paper is organized as follows: section 1.1 presents background information on the rhythmic structure in Carnatic music while section 2 presents previous research on automatic music generation methods. Section 3 describes the proposed approach while section 4 discusses the evaluation of the method. Discussion and Conclusions are drawn in sections 5 and 6 respectively.

1.1 Rhythmic structure in Carnatic music

The rhythmic framework of Carnatic music is based on the tāla, which provides a structure for repetition, grouping and improvisation. The tāla consists of a fixed time length cycle called āvartana, which can also be called the tāla cycle. The āvartana is divided into equidistant basic time units called akṣaras, and the first akṣara of each āvartana is called the sama [3]. Two primary percussion accompaniments in Carnatic music are the Mridangam and Kanjira. The Mridangam is made of a cylindrical shell with stretched membranes on either side of the in-

strument body. While one side is loaded with a black paste that creates a pitched tone, the other membrane creates a bass-like sound. The Kanjira on the other hand is a frame-drum, with a tonally rich membrane. Unlike the Mridangam, the Kanjira is not tuned to a specific key, but it can cover a wide range of frequencies with especially rich lower frequencies. The rhythmic complexities of Carnatic rhythm are especially showcased during the solo or taniavartanam. First, each instrument performs separately and then they trade off in shorter cycles with a precise question-answer like session, followed by a joint climactic ending. All training excerpts used in the proposed generation method were performed by the Kanjira drum in the context of a concert solo. We decided to use the Kanjra compositions as a training corpus because the strokes had a simpler frequency distribution compared to the Mridangam.

2. RELATED WORK

Probably the most popular study of musical style imitation is David Cope's Experiments in Musical Intelligence (EMI) system. EMI analyzes the score of MIDI sequences in terms of patterns and stores the patterns in a database where the system learns the style of a composer given a number of training examples [4]. Bel and Kippen [5] present the Bol Processor, a software system that models tabla drumming improvisation. The system is based on a linguistic model derived from pattern languages and a formal grammar that has the ability to handle complex structures by using a set of training examples. Dias and Guedes in [1] discuss a contour based algorithm for real time automatic generation of jazz walking bass lines, following a given harmonic progression. The algorithm generates melodic phrases that connect the chords in a previously defined harmonic grid, by calculating a path from the current chord to the next, according to user-defined settings controlling the direction and range of the melodic contour. Biles in [2] developed a generative system for composing jazz solos based on a genetic algorithm, which starts with some initial musical data initialized randomly or by human input. Using a repeated process similar to biological generation the system produces similar musical data. Dias et al [6] present the GimmeDaBlues app that allows the user to play jazz keyboard and solo instruments along a predefined harmonic progression, by automatically generating the bass and drums parts, responding to the user's activity. Assayag, Dubnov and Delerue [7] proposed a dictionary based universal prediction algorithm that provides an approach to machine learning in the domain of musical style. Operations such as improvisation or assistance to composition can be realized on the resulting representations. The system uses two dictionaries, the motif and continuation. A generation algorithm is used to predict a sequence based on the motif dictionary. The continuation dictionary gives probabilities of various continuations and is used to determine the next symbol. Pachet discusses the continuator [8] an interactive imitation system, which generates new melodic phrases in any style, either in standalone mode or as continuations of musician's input. The system is based on an incremental parsing algorithm to train a variable-length Markov chain that stores possible probabilities of sequences. The system progressively learns new phrases from a musician and develops a robust representation of his or her style. A framework for generating similar variations of guitar and bass melodies is proposed in [9]. The melody is initially segmented into sequences of notes using onset detection and pitch estimation. A set of hierarchical representations of the melody is estimated by clustering the pitch values. The pitch clusters and the metrical locations are then used to train a prediction model using variable-length Markov chain.

3. SYSTEM IMPLEMENTATION

3.1 Dataset

The training corpus consisted of 8 percussion solo compositions in *ādi* tāla (8 beat- cycle) in three different tempo levels (slow/moderate/fast). The compositions were performed by Akshay Anantapadmanabhan, in the Kanjira. These examples were recorded using a metronome.

All excerpts were manually annotated using Sonic Visualizer [10] including the sāmā and the other beats comprising the tāla. Each stroke event was coded as a string based on its register (Lo-Mid-Hi), the inter-onset-interval (IOI) between strokes and a value indicating the velocity of the stroke. The fourth author annotated the music excerpts by using the following process: The metronome was recorded in a separate channel and used as reference for each performance. A note onset transformation was estimated for the audio track by which note onsets were detected. By looking at the note onsets, the spectrogram of the sound and by listening to it at a reduced playback speed, the different types of strokes were categorized into three categories and the annotation marker positions were manually adjusted. Based on the spectrogram analysis, the frequency spectrum of the strokes was divided into three frequency bands (low, mid and high) depending on the frequency content of each stroke (110-190 Hz for low, 190-600 Hz for mid and 600-1200 Hz for high strokes). Although the Kanjira has a richer variety of registers and strokes, the reduction to three registers was a step to simplify the different stroke definition. This reduction was validated by Anantapadmanabhan as a process to faithfully encode the different strokes in the Kanjira**.** The normalized velocity values of the strokes were obtained by computing an onset detection function, and estimating its amplitude level with a value between 0.2 and 1 according to the strength of the stroke. In the present work, the complex domain onset detection [11] was used to compute the onset detection function implemented in the Vamp-plugins in version of Sonic Visualizer. Table 1 lists the coded feature values used to model each stroke event.

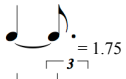
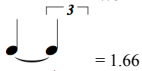
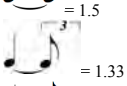
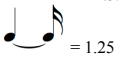
Features	Value
Register	Lo-Mid-Hi
IOI duration (sec)	T1  = 2 T2  = 1.75 T3  = 1.66 T4  = 1.5 T5  = 1.33 T6  = 1.25 T7  = 1 T8  = 0.75 T9  = 0.66 T10  = 0.5 T11  = 0.33 T12  = 0.25 T13  = 0.16 T14  = 0.125
Velocity	V1 (0.2) V2 (0.5) V3 (0.8)

Table 1. Features for modeling stroke events.

3.2 N-gram model

All coded stroke events from the compositions were merged in a single training corpus to learn a statistical model. We used n-gram analysis to model the underlying rhythmic progression of the strokes in the training data. The general n-gram definition is given in (1), while the representations of a unigram, bigram and trigram are given in (2), where *s* denotes a stroke event.

$$p(s_i \mid s_1,\ldots,s_{i-1}) = p(s_i \mid s_{i-n+1},\ldots,s_{i-1}) \tag{1}$$

$$\begin{aligned} &unigram: && p(s_i) \\ &bigram: && p(s_i \mid s_{i-1}) \\ &trigram: && p(s_i \mid s_{i-2}, s_{i-1}) \end{aligned} \tag{2}$$

An example of a trigram encoding the strokes is given below:

MidT10V1 LoT12V3 LoT12V3

This trigram consists of three stroke events. The first stroke has a Mid register with an eighth note duration performed with 0.2 velocity followed by two strokes with Lo register and sixteenth note duration performed with 0.8 velocity.

We estimated the n-gram probabilities up to a five-gram by counting the frequency of the strokes on the training corpus where *N* is the total numbers of stroke events in the training data. The unigram and bigram probabilities are calculated using equations (3) and (4) where s_a denotes a particular stroke event, s_b its preceding stroke and c the count of a stroke:

$$\hat{p}(s_a) = \frac{c(s_a)}{N} \tag{3}$$

$$\hat{p}(s_b \mid s_a) = \frac{c(s_a, s_b)}{\Sigma_{sb} c(s_a, s_b)} \approx \frac{c(s_a, s_b)}{c(s_a)} \tag{4}$$

The n-gram model provides the transition probabilities between stroke events. For example, consider the case of a bigram model where two stroke events are present. The first tagged as Mid stroke register with a quarter note duration and with 0.2 velocity value and the second as a Lo stroke register with a sixteen note duration and velocity value of 0.2. What would the probability be that the next stroke will be a Lo stroke register with a sixteen note duration and 0.8 velocity given the previous stroke events representation?

We computed all the n-grams probabilities up to a five-gram because we wanted to test how past information and size of accumulated memory could affect and change the generation process. All n-gram probabilities were stored in tables to be used later during the generation process. The generation process used these data to generate new strokes events sequentially. Given a sequence of strokes, a stroke event is generated based on the weighting probability of the most likely stroke to follow given the previous strokes.

3.3 Generation

The generation process depends on the n-gram selection of and on the number of stroke events. If a trigram is selected the generation starts with the first trigram of the training file. The next stroke event is generated based on the probabilities of trigrams that start with the last two stroke events in the generated sequence. When the stroke event is generated the algorithm looks for the next last two stroke events in the sequence to generate the next stroke and search again for the highest probability of trigrams that start with the last two stroke events. This process is iterative until the number of initial selected stroke events is reached. The overall process is presented in Figure 1.

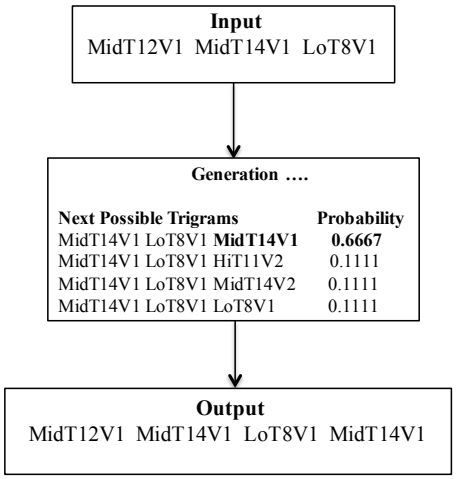


Figure 1. Generation process using a trigram model and probability estimation.

4. EVALUATION

Several approaches for the evaluation of generative music systems have been proposed in the past. Researchers have tried to use Turing tests [12] to compare the output between computer-aided and non-computer aided compositions by measuring the degree of perceptual quality. This model has been criticized in the past for its application in executing and evaluating listener surveys [13]. Pearce and Wiggins [14] use a set of musical examples to train a genetic-algorithm based system. A discrimination test is used to evaluate whether the output of the system can be distinguished from the training compositions. Cont, Dubnov and Assayg [15] evaluate a generative system using the same model as a classifier. The model is trained for a particular style of music and outputs a probability to a given music excerpt. A quasi-Turing test is used in [5] to evaluate the Continuator. The evaluation is used to assess to what extend a listener can determine that a melody generated by the system was composed or played by a human or by a machine. Collins [16] evaluates an algorithmic system that creates electroacoustic art music by using three expert composer judges. They evaluate the system based on how music material was assembled its form, structure and instrumentation. The authors in [9] use a group of experts to evaluate an automatic guitar and bass phrase continuation melody system. Their feedback is related to the type of similarities and differences they notice between the original and generated examples and the aesthetic outcome.

Since we are interested in generating new sequences of Carnatic music percussion from the training data and there is no benchmark dataset for music generation performance of other systems we decided to conduct a preliminary evaluation based on the feedback of the musician who also provided the dataset. The fact that Anantapadmanabham is an expert in Carnatic music percussion and also provided the dataset that we used for analysis provides a unique set of conditions to do a preliminary evaluation of the generative model and this approach.

A questionnaire was prepared and presented to Anantapadmanabhan. The new sequences were generated using different n-grams (bigram, trigram, fourgram and fivegram) with duration of 2 minutes each. He was asked first to listen to the compositions as many times needed to get familiar with the rhythmic structure and development of the excerpts and then answer the questionnaire.
Examples of the generated excerpts can be downloaded at https://github.com/Trochidis/CAMeL-Carnatic-Percussion-Music-Generation-Using-N-Gram-Models.
Anantapadmanabhan was first asked to judge if the generated compositions contained recognizable Carnatic music rhythmic patterns, which he positively answered. The next question of the form was related to the short-term level of rhythmic structure asking if the rhythmic patterns were occurring in metrical appropriate positions. He answered that sometimes they were and others they were not. Based on his feedback there were certain strokes, particularly in the percussive roll sections of the generated compositions that they were repeated consecutively. This sometimes created a feeling that the same succession of strokes kept playing without variation which does not usually happen in the rhythmic structure of Carnatic music. The next question was related to the long-term evolution and rhythmic progression asking if the rhythmic structure of the generations evolved in time as expected in this style. He answered that most of the generated compositions in particular the ones with the shorter memory (bigram-trigram) failed to capture the long-term rhythmic structure and the correct transition between longer rhythmic structures.
His additional comments were that n-grams with larger memory such as fourgrams and fivegrams were more successful in capturing Carnatic rhythm groupings compared to bigrams or trigrams and contained more rhythmic patterns in resemblance with the original Carnatic music patterns.

5. DISCUSSION

This work presents a method for automatically generating new Carnatic style rhythmic patterns based on a set of training examples. An n-gram analysis and Markov Chains are used to model short and long-term patterns and represent rhythmic progressions. Based on the expert's feedback the method is able to generate recognizable Carnatic-style rhythmic patterns with some success. The evaluation indicates that the n-gram analysis is more successful on capturing short-term rhythmic patterns compared to long-term ones. Moreover, larger n-grams such as fourgrams or fivegrams generate more appropriate and interesting Carnatic style rhythmic patterns. This is due to the fact that they are more successful in modeling the long-term pattern transitions compared to shorter structures such as unigrams and bigrams. An improvement over the current method will be to implement a cluster analysis to the larger n-grams i.e fivegrams or sevengrams and generate rhythmic patterns based on the cluster transitional probabilities. This might improve the rhythmic representation and progression of long-term rhythmic structures compared to the one implemented in our current system.

Another approach to tackle the problem of long-term representation of rhythmic progression is to use a Long Short Term Memory Recurrent Neural Network (LSTM-RNN) [17]. LSTM RNNs are capable of learning long-term dependencies and have been used successfully in language modeling and speech recognition. The LSTM RNNs architecture is based on a dynamic memory with cells that stores information about the previous states. They can combine previous states and current memory to make decisions and efficiently capture long-term dependencies by dynamically changing their memory.

6. CONCLUSION

In the present paper, a method for automatically generating Carnatic style rhythmic patterns is explored. By extracting features such as the stroke register (Lo-Mid-Hi), inter-onset interval duration of the strokes and amplitude of the strokes the system is capable of automatically generating new rhythmic progressions stylistically similar to the training compositions. N-gram analysis and statistical learning is used to model the rhythmic structure and build the rhythmic development using the extracted features. The generated outcome was evaluated by a professional composer and percussionist of Carnatic music in terms of rhythmic development and musical aesthetics. Feedback from the evaluation shows that the method is capable of generating new interesting Carnatic style rhythmic patterns by training on previous data. Future work will test the method on a larger dataset of recordings and evaluate the effectiveness of the method by conducting a perceptual study using a group of professional Carnatic musicians. Furthermore, we would like to perform a statistical analysis of the evaluation results to test the percentage and the strength of the generated excerpts that were positively evaluated by the human-experts. Finally, we aim to test the method against other approaches such as clustering and deep belief networks.

7. REFERENCES

[1] R. Dias, C. Guedes, “A Contour-Based Jazz Walking Bass Generator.” Proceedings of the Sound and Music Computing Conference, 2013.

[2] J.A. Biles, “GenJam in Transition: from Genetic Jammerto Generative Jammer”, International Conference on Generative Art, Milan, 2002

[3] P. Sambamoorthy, South Indian Music Vol. I-VI, The Indian Music Publishing House, 1998.

[4] D. Cope, Experiments in musical intelligence (Vol. 12). Madison, WI: AR editions, 1996.

[5] B. Bel & J.Kippen. Modelling music with grammas: formal language representation in the Bol Processor. Computer Representations and Models in Music, Academic Press, pp.207-238, 1992

[6] R. Dias, T. Marques, G. Sioros and C. Guedes, “GimmeDaBlues: an intelligent Jazz/Blues player and comping generator for iOS devices”. in Proc. Conf. Computer Music and Music Retrieval (CMMR 2012), London 2012.

[7] G. Assayag, S. Dubnov, & O. Delerue. “Guessing the composer’s mind: Applying universal prediction to musical style”, In Proceedings of the International Computer Music Conference, 1999, (pp. 496-499).

[8] F. Pachet. “The continuator: Musical interaction with style”, in J. New Music Research, 2003, 32(3), 333-341.

[9] S. Cherla, H. Purwins, & M. Marchini, “Automatic phrase continuation from guitar and bass guitar melodies”, in Computer Music Journal, 2013, 37(3), 68-81.

[10] C. Cannam, C. Landone, & M. Sandler, “Sonic visualiser: An open source application for viewing, analysing, and annotating music audio files”, in Proc. Int. Conf. on Multimedia, 2010, (pp. 1467-1468). ACM.

[11] C. Duxbury, J.P Bello, M. Davies & M. Sandler, “Complex domain onset detection for musical signals”, in Proc. Digital Audio Effects Workshop (DAFx), 2003, (No. 1, pp. 6-9).

[12] A. M. Turing, “Computing machinery and intelligence”. Mind, 1950, 59(236), 433-460.

[13] C. Ariza, “The interrogator as critic: The turing test and the evaluation of generative music systems”, Computer Music Journal, 2009, 33(2), 48-70.

[14] M. Pearce, and G. Wiggins, “Towards a Framework for the Evaluation of Machine Compositions”, in Proc. of the AISB01 Symposium on Artificial Intelligence and Creativity in the Arts and Sciences. Brighton, UK: SSAISB, 2001, pp. 22–32

[15] A. Cont, S. Dubnov, & G. Assayag, “Anticipatory Individual Behavior-Anticipatory Model of Musical Style Imitation Using Collaborative and Competitive Reinforcement Learning”, Lecture Notes in Computer Science, 2007, 4520, 285-306.

[16] N. Collins, “Automatic composition of electroacoustic art music utilizing machine listening”, Computer Music Journal, 2012, 36(3), 8-23.

[17] N. Boulanger-Lewandowski, Y. Bengio and P. Vincent, “Modeling Temporal Dependencies in High-Dimensional Sequences: Application to Polyphonic Music Generation and Transcription”, In Proceedings of the 29th International Conference on Machine Learning (ICML), 2012

PROTOTYPING A WIRELESS INTEGRATED WEARABLE INTERACTIVE MUSIC SYSTEM:MUSFIT

Yu-Chung Tseng **Bo-Ting Li** **Tsung-Hua Wang**
Master Program of Sound and Music
Innovative Technologies
National Chiao Tung University
eamusic.tseng@msa.hinet.net lbtnmpcm@hotmail.com lovemine@gmail.com

ABSTRACT

This paper presents the development of a wireless integrated wearable interactive music system - Musfit. The system was built according the intension of integrating the motion of hands (fingers), head, and feet of a performer to music performance. The device of the system consists of a pair of gloves, a pair of shoes, and a cap, which were embedded various sensors to detect the body motion of a performer. The data from detecting was transmitted to computer via wireless device and then mapped into various parameters of sound effectors built on Max/MSP for interactive music performance.

The ultimate goal of the system is to free the performing space of the player, to increase technological transparency of performing and, as a result, to promote the interests of interactive music performance.

At the present stage, the progression of prototyping a wireless integrated wearable interactive music system has reached the goal we expected. Further studies are needed in order to assess and improve playability and stability of the system, in such a way that it can be effectively employed in concerts eventually.

1. INTRODUCTION

Wearable devices with sensors embedded have been widely adopted in human-computer interaction and new interfaces for musical expression communities [1]. A wireless wearable interactive music device controlled by body posture has been increasingly developed. A known example of wireless wearable devices for interactive music performance is Laetitia Sonami's Lady's Glove built and developed by Stein [2].

Different from Lady's Glove, the wireless wearable interactive music system - Musfit was intended to integrate the motion of hands (fingers), head, and feet of a performer to music performance.

The device of the system consists of a pair of gloves, a pair of shoes, and a cap. Several hardware including Arduino [3], Bluetooth, and various sensors were employed in the system. Sensors built in gloves, shoes, and a cap, were used to detect the motion of performer's feet, fingers, and head.

The data from detecting was transmitted into computer via Bluetooth device and was mapped into various parameters of sound effectors via the algorithms of Max program. In addition, the mapped numbers were also used to trigger the mode switch of sound effectors or to trigger the mode of sound diffusion.

The object of the research is to prototype a wireless integrated wearable interactive music system, which could free the performing space of the player and increase the technological transparency of performance while involving the use of technologies in music concert. Figure 1 shows the prototyped wireless integrated wearable interactive music system - Musfit.

Figure 1. Musfit, a prototyped wireless integrated wearable interactive music system

2. STRUCTURE OF SYSTEM

In this section, the structure of interactive music system, structure of hardware and structure of software are described.

2.1 Structure of Interactive Music System

In the system, sensors built in the cap, gloves and shoes will detect performer's body motion. Sensor data from motion detection was transmitted into computer and was mapped into various parameters of sound effectors via algorithms of Max/MSP to transform the performer's voice collected from microphone. In addition, the data was also used to trigger the mode of sound effectors or mode of sound diffusion in a music performance.

For a performer, Musfit allows performer interacts with the system by body motion in real time to achieve her/his desired sounds or music results. For audience, the association of audio output from speakers and visual element from watching the body motion of the performer promotes the interests of appreciation of an interactive music performance. Figure 2 shows the overall structure of Musfit.

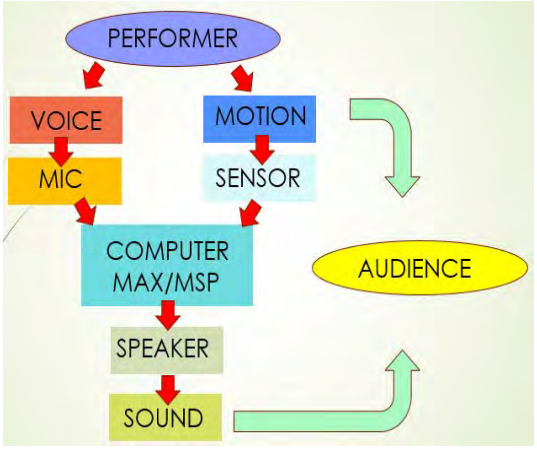


Figure 2. Overall structure of Musfit.

2.2 Structure of Hardware System

The hardware of the system was built on Arduino Nano. The PCB was employed to integrate the power supply, Arduino, Bluetooth, and various sensors. Figure 3 shows the structure of hardware of Musfit. Figure 4 shows the assembly of various hardware inside the gloves (same as those in shoes and cap)

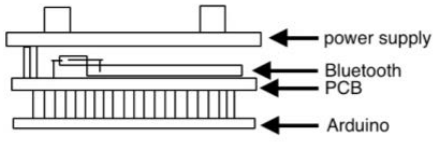


Figure 3. Structure of hardware system of Musfit.

Figure 4. Assembly of hardware inside the glove.

Based upon the structure of hardware, sensor data from the body motion detection was converted into digits by Arduino, which were then transmitted into computer via Bluetooth device. Figure 5 shows the flow of data in the hardware system.

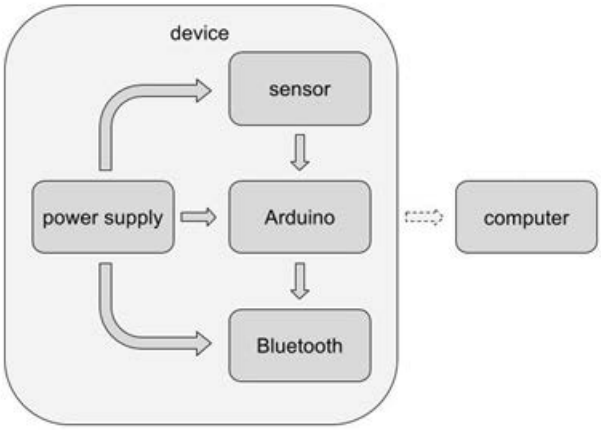


Figure 5. Flow of data in hardware system.

2.3 Structure of Software System

The software used for the interactive music system was based on the object-oriented music program-Max/MSP. The structure of software system consists of three main sections:

1. Input Section, which receives the Audio Source from microphone and Bluetooth parameters from serial port. 2. Signal Process Section, which consists of 2 sets of sound effectors: FX1 (ORI (Original), RM (Ring Modulation), and HARM (Harmonization)) and FX2 (Gran (Granular Synthesis), REV (Reverb), and FB (Feedback)). Those effectors process or modulate the input sound source according the dada received from Bluetooth. 3. Signal Output Section, which consists of 2 modes of sound output- stereo and 4-channel. Figure 6 shows the structure of software system of Musfit.

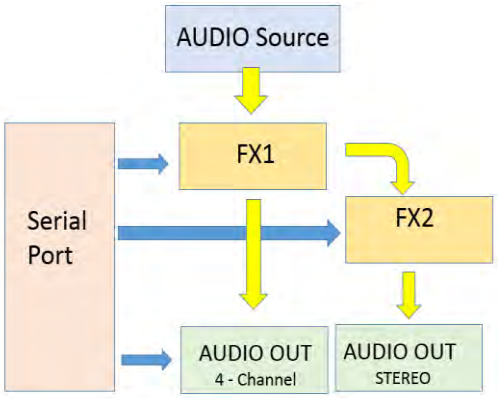


Figure 6. Structure of software system of Musfit.

3. DESIGN AND IMPLEMENT

In this section, the design and implementation of the device, including the cap, the gloves, and the shoes, and of software programming of the system are described.

3.1 Device

The electric compass was built on the brim of a cap to detect the motion angles of head. Figure 7 shows the installation of GY-273 electric compass on a cap.

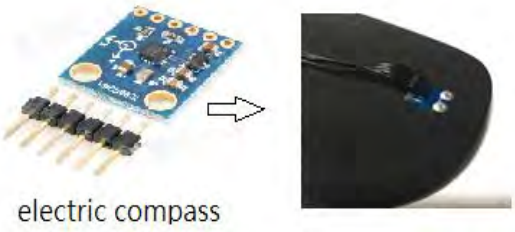


Figure 7. GY-273 electric compass on a cap.

Three bend sensors were built on index, middle, and ring fingers of each glove to detect the bending degree of fingers. Figure 8 shows the installation of bend sensors in the gloves on index, middle, and ring fingers as indicated by arrows.

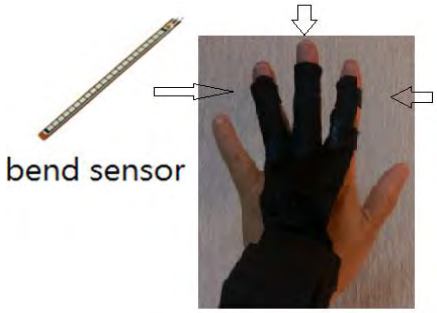


Figure 8. Bend sensors on fingers of a glove.

Pressure sensors were built in the front and rear insoles of shoes to detect the weights of feet stepping. Figure 9 shows the installation of pressure sensors on an insole.

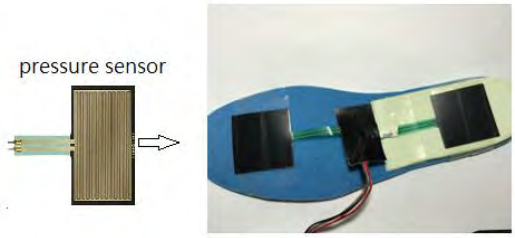


Figure 9. Pressure sensors on an insole.

3.2 Software Programming

In the following texts, the design and implementation of the software programming are described.

3.2.1 Main User Interface

For a wearable music performance, a user friendly interactive interface is crucially needed to allow performer easily to monitor the change of parameters on screen and interact with the system to control the desired music results in real time.

Built on Max/MSP program environment, the main user interface of the system providers the performer four main sections of interactive information separated by different colors displayed in a presentation mode of the program. The information includes data information, audio information of left and right hand(including ORI, RM, Harm, Gran, Rev, and FB), select of sound effectors modular, and modes of sound output(either stereo or 4-channel). Figure 10 shows the main interactive interface of Musfit based on Max/MSP.

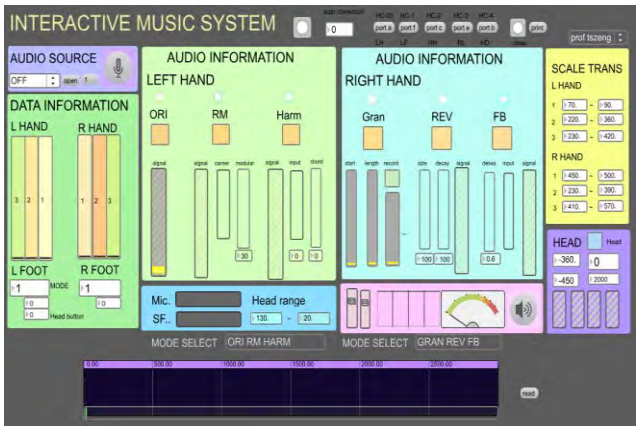


Figure 10. Main interactive interface of Musfit based on Max/MSP.

3.2.2 Sound Effector Modular Design

Since there were several sound effectors employed in the system, a design of preset of different sound effectors was also concerned. Sound effector modular allows performer to quickly select effector setting before the performance. Figure 11 shows the preset of sound effector modular of left hand.

Figure 11. Preset of sound effector modular of left hand.

4. MAPPING

The importance of mapping between gestures and sound is an active field of research [4] [5]. In this section, ,as shown on Table 1, the range settings and limits of sensors, the range of value for mapping, and the mapping associated with each part of the wearable device or motion of fingers, hand, and feet to various parameters of sound controls are described.

	Head (cap)	**Hand (glove)**	**Feet (shoe)**
Range settings and limits of sensors	Compass: North Gyroscope ： 2 axes	Bending degree： 0～120	Weight range: 20g～2kg
Range of value	Heading : 0 ~ 360 X、Y : -360 ~ 360	0 ~ 1023	0 ~ 1023
Value to Sound Control Mapping	Spatilization of sound	Parameters of effectors	Mode switch of effectors and output channel

Table 1. Range of value and mapping of various values to sound controls.

4.1 Hand and Fingers (Gloves)

In the system, each hand was assigned to control different sound effectors. The left hand was assigned three types of sound effectors, including Original (without effect), Ring modulation, and Granular Synthesis. The right hand was assigned another three types of sound effectors, including Panning, Feedback, and Reverb. Figure 12 shows the assignments of controls of left and right hand.

Figure 12. Assignments of controls of left and right hand.

In the system, Fingers were used to control sound effectors via bend sensors built in the glove, which detect the bending degree (ranging from 0 to 120) of fingers. The data was converted into digits (ranging from 0 to 1023) by Arduino, and was then rescaled and mapped into parameters of effectors via algorithms of Max program.

Fingers of both hands play different roles in controlling. Fingers of left hand were used to control different parameters of various sound effectors. For example, index, middle, and ring finger were used to control the modular frequency, the carrier signal, and the output amount of signal of ring modulation respectively; index, middle, and ring finger were used to control the chord change, the carrier signal, and the output amount of signal of granular synthesis respectively. Table 2 shows the mapping of fingers to parameters of sound effectors.

Mode/Left	**Finger**	**Description**
Original	Non	Bypass original
Ring Modulation	Index	Modular frequency
	Middle	Carrier signal
	Ring	Signal amount
Granular synthesis	Index	Chord change
	Middle	Carrier signal
	Ring	Signal amount

Table 2. Mapping of fingers of left hand to parameters of sound effectors.

Giving more examples, index, middle, and ring finger of right hand were used to control the Granular start point, the Grain length, and Record switch of Granular synthesis respectively; index, middle, and ring finger of right hand were used to control the delay volume, the delay time, and the output amount of signal of Feedback respectively. Table 3 shows the mapping of fingers of right hand to various parameters of different sound effectors.

Mode/Right	Finger	Description
Granular synthesis	Index	Granular start
	Middle	Grain length
	Ring	Record switch
Feedback	Index	Delay volume
	Middle	Delay time
	Ring	Signal amount
Reverb	Index	Reverb size
	Middle	Decay time
	Ring	Signal amount

Table 3. Mapping of fingers of right hand to parameters of sound effectors.

4.2 Feet (Shoes)

Feet were used to control the mode switch of sound effectors and mode of 4-channel output via the pressure sensors, which were built on the insoles inside the shoes to detect the weights of feet stepping (ranging from 20g-2kg). The data from detecting was converted into digits by Arduino (ranging from 0 to 1023) and then mapped into parameters via algorithms of Max. When a specific parameter reaches the preset threshold, the mode switch of sound effectors and mode of 4-channel output were triggered; parameters from heel sensor of left and right foot trigger the mode switch of sound effectors; parameters from the forefoot sensor of left foot and heel sensor of right foot trigger the 4-channel sound output mode respectively. Figure 13 shows the mapping of feet to mode and 4-channel output trigger

Figure 13. Mapping of feet to mode and 4-channel output trigger.

4.3 Head (Cap)

Head was used to control the sound spatialization via the electric compass (with gyroscope), which was built on the brim of a cap to detect the moving angle of head motion both horizontally and vertically. After converting, the data of horizontal angle (ranging from 0 to 360) was used to control the sound movement among 4 speakers; data of vertical angle was used to decide the decay time of sound in a speaker. Figure 14 shows maping of head motion to 4-channel sound spatialization control and decay time of sound (in a speaker).

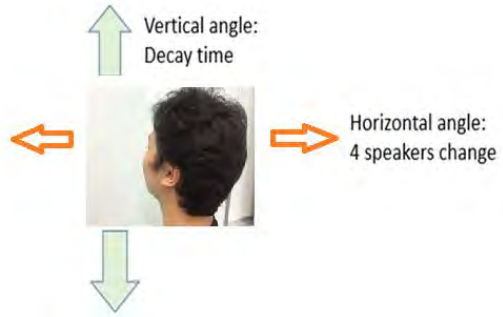


Figure 14. Maping of head motion to 4-channel sound spatialization control and decay time of sound (in a speaker).

5. CONCLUSION

At the current stage, the development of Musfit has reached the goal we set: prototyping a wireless integrated wearable interactive music system. However, further studies are needed in order to assess and improve playability and stability of the system, in such a way that Musfit can be effectively employed by performer in concerts.

We hope, in in near future, an optimized version of Musfit could be created, which could not only truly free the performing space of the musician, but also could increase the technological transparency of performance to help audience realize the relationship between music and technology and, as a result, to promote the interests of appreciating an interactive music performance.

Acknowledgments

This research was supported by the Ministry of Science and Technology in Taiwan (R.O.C.). (Project No. 104WFA0650423)

6. REFERENCES

[1] D. J. Sturman and D. Zeltzer. A survey of glove-based input. Computer Graphics and Applications, IEEE, 14(1):30–39, 1994.

[2] Lady's Glove. http://www.sonami.net/lady_glove2.htm

[3] Arduino. http://www.arduino.cc.

[4] T.Winkler. Making motion musical: Gesture mapping strategies for interactive computer music. In ICMC Proceedings, pages 261–264, 1995.

[5] R. I. Godøy and M.Leman.Musical gestures: Sound, movement, and meaning. Routledge, 2009.

THE HYPER-ZAMPOGNA

Luca Turchet
Department of Automatic Control
KTH Royal Institute of Technology
turchet@kth.se

ABSTRACT

This paper describes a design for the Hyper-Zampogna, which is the augmentation of the traditional Italian zampogna bagpipe. The augmentation consists of the enhancement of the acoustic instrument with various microphones used to track the sound emission of the various pipes, different types of sensors used to track some of the player's gestures, as well as novel types of real-time control of digital effects. The placing of the added technology is not a hindrance to the acoustic use of the instrument and is conveniently located. Audio and sensors data processing is accomplished by an application coded in Max/MSP and running on an external computer. Such an application also allows for the use of the instrument as a controller for digital audio workstations. On the one hand, the rationale behind the development of such augmented instrument was to provide electro-acoustic zampogna performers with an interface capable of achieving novel types of musical expression without disrupting the natural interaction with the traditional instrument. On the other hand, this research aimed to provide composers with a new instrument enabling the exploration of novel pathways for musical creation.

1. INTRODUCTION

The last decades have seen an increasing interest towards the development of conventional acoustic instruments enhanced with sensor technology and digital signal processing techniques. These instruments are usually called "hyper instruments" [1] or "augmented instruments" [2], and are conceived to extend the sonic possibilities offered by the instrument in its original version. The performer's interactions with the sensors are used to control the production of the electronically generated sounds that complement, or modulate, the sounds acoustically generated by the instrument.

Some principles for the design of such new musical interfaces have been proposed [3–5]. In addition research has also focused on the importance of mapping strategies between the player's gestures and the controlled sound parameters [6–8], which have an important impact on how the instrument will be played and on how the audience will perceive the performance.

Quite a number of augmented instruments have been developed. However, looking at papers written on traditional instruments augmentation the number is quite low. Examples of augmented traditional instruments are the "electronic sitar controller" [9], the "hyperpuja" [10], or the "hyper-hurdy-gurdy" [11]). To the author's best knowledge no research has been conducted yet on the acoustic augmentation of one of the most typical exemplars of traditional instruments: the bagpipe [12].

Nevertheless, a number of people from both academy and industry have worked on the application of electronics to various types of bagpipe. Among commercially available solutions, one can cite the electronic bagpipes produced by TechnoPipes [1] , DegerPipes [2] , Master Gaita [3] , Redpipes [4] . Typically, these fully electronic instruments consist of a chanter-like interface where single capacitive touch-switches are used in place of the tone-holes, which act as MIDI controller and/or a controller for a bagpipe sound synthesizer (usually involving wavetable synthesis).

Within the academic community there have been various efforts to improve the expressive capabilities of such controllers. Indeed usually these devices are not characterized by an accurate tracking of the partial occlusions of the tone-holes, which are typically involved in the acoustic instrument to slide between notes. The EpipE is an ad-hoc built chanter interface based on the Irish Uilleann Pipes [13, 14] and is used as a MIDI controller. It is characterized by an array of sixteen small binary touch-switches for each tone-hole that enable the sensing of various degrees of tone-holes coverage. The instrument is also equipped with a force-sensitive resistors that allows for the measurement of the pressure exerted on the bag by the player's arm. These features allows for the mimicking the feel and responsiveness of the corresponding acoustic instrument. The FrankenPipe is a MIDI controller consisting of an acoustic chanter of a great highland bagpipe that is enhanced with photoresistors placed underneath each hole, and of an air-pressure sensor deployed in the bag [15]. These features allow a player to maintain the physical feel of playing the traditional instrument. In a different vein, more recently, an electronic bagpipe chanter interface and software system has been developed to assist in the process of learning the great highland bagpipe [16–18]. The technology involved in the chanter consists of infrared reflectance sensors, which serve the purpose to de-

[1] http://www.fagerstrom.com/technopipes
[2] http://www.deger.com
[3] http://mastergaita.com
[4] http://redpipes.eu

tect the continuous movements of the player's fingers, and an air pressure sensor, which is used in place of the chanter reed and allows the chanter to be connected to a traditional acoustic set of pipes.

This paper presents the augmentation of a bagpipe typical of the Italian musical tradition: the zampogna. This instrument (described more fully in Section 2) has a long and strong tradition in various Italian regions [19], each of which has given rise a particular model of it. This type of bagpipe has been already object of the interest of researchers in the sound and music computing community. The e-Zampognë is an interface based on the zampogna that is used as a controller for the sound synthesis of various zampogna models [20]. The uniqueness of e-Zampognë lies in the fact that is the sole interface involving double or triple-chanter, while the other systems described above are based on a single chanter. However, all the reviewed systems based on the various bagpipes models have not faced the challenge of augmenting the sonic possibilities of a bagpipe while preserving its original acoustic sound: they are just controller interfaces for sound synthesis.

This research was motivated by the author's artistic need to make the zampogna an interface capable of enabling musical expressions neither achievable with the traditional instrument nor with the application to it of current commercially available technologies for sound processing, nor with the interfaces of current electronic bagpipes. In Section 2 a brief description of the zampogna is provided to make this paper more intelligible to those unfamiliar with the instrument.

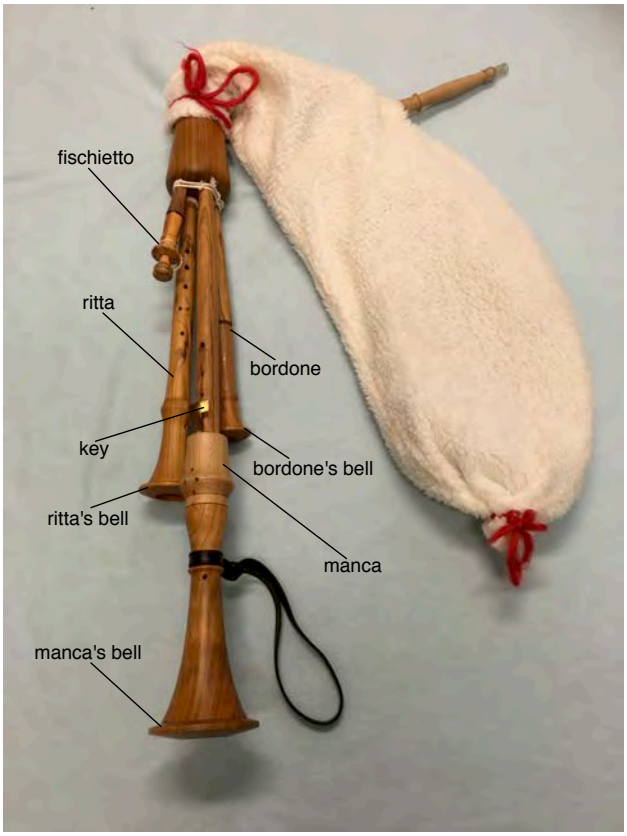


Figure 1. An exemplar of zampogna bagpipe with the indication of its main components.

2. ZAMPOGNA DESCRIPTION

The zampogna is a bagpipe typical of the central and southern part of Italy [19]. It has a bright and powerful sound very rich in overtones. There are many types of zampogna bagpipes, with the main differences being timbre, tuning, size, number of pipes, and types of materials used. The type of zampogna object of this research is the so-called "zampogna a chiave" (literally zampogna with key), which belongs to the most pure tradition of the Molise region of Italy. Figure 1 illustrates an exemplar of zampogna a chiave crafted by master Luigi Ricci from Scapoli. This particular model of zampogna is identified by four unequal pipes, two chanters and two drones, all with double reeds and all ending with a conical bell to emphasize the sound. The first chanter is a soprano chanter called "ritta" and is played by the fingers of the right hand. It has nine tone-holes but only five are actually used by the hand. The second chanter is a long bass chanter called "manca" and is played by the fingers of the left hand with the exception of the thumb. It has four tone-holes and the last one is covered with a metallic key (from which the epithet "a chiave"). The first drone is called "bordone" and its tuning is changed by the thumb of the left hand. The second drone (not always present in the various zampogna models), is the smallest pipe and is called "fischietto" (literally little whistle). Its tuning is not changeable and can be activated or deactivated thanks to a detachable cap.

Like most of bagpipes, the air supply is achieved by means of a bag (traditionally made of animal skin, e.g., got, but currently mostly with synthetic material, e.g., Gore-Tex) that is held under the player's right arm. The player ensures a steady flow of air through reeds of all pipes by maintaining a constant pressure on the bag with the elbow. As with all bagpipes, it is worth noting that the instrument does not afford a real control on the dynamics. The main difference between zampogna and other bagpipes lies in its polyphonic capabilities due to the number of pipes for melody greater than the usual one. In addition, unlike the vast majority of bagpipes, all the pipes in the zampogna are planted in the same wood block connected to the bag. Regarding tone-holes effects, zampogna allows for partial occlusions: the tone-holes of ritta, manca, and bordone can be gradually covered and uncovered to slide between notes.

3. DESIGN

The design of the augmentation of the zampogna bagpipe originated from the results of a long-lasting research on how to extend the sonic possibilities of the instrument and overcoming its limitations when used in conjunction with the most widespread current technologies for sound processing. Such a research was entirely based on the author's personal needs as a performer to avail himself of a novel interface for musical expression, capable of opening unexplored paths for composition, improvisation, and performance. These needs led to the following requirements that guided the design:

- The added hardware technology should have been easy to put on and remove, and the instrument could have been still played in the normal acoustic way. This resulted in the design choice of enhancing the instrument without physically modifying it with holes or carvings;
- The augmentation should have kept unaltered all the conventional set of gestures to play the instrument. This resulted in the minimization of the amount of technology, in its hiding as much as possible from the player's fingers, and by adopting wireless solutions. This also led to the identification of the possible set of new gestures that a performer would act on the instrument without interfering with the natural act of playing;
- The hardware and software technology should have supported the separate tracking and consequent independent modulation of the various pipes;
- The hardware and software technology should have allowed zampogna players to achieve unprecedented musical expressions such as sound modulations, sound spatialization, and generation of additional synthesized sounds.

Considering the set of requirements listed above, the research conducted during the design phase focused on the identification of new possible and reasonable set of gestures that could be added to the normal playing technique, the selection of the types of sensors to track such gestures, the identification of the positions where placing the selected sensors, and on the definition of mapping strategies between the player's gestures and the sound production.

3.1 Interaction design

The design of novel performer-instrument interactions started with the accurate analysis of the zampogna playing technique. This requires to have basically always both hands on the instruments not only to keep the fingering position, but also to sustain the instrument. Therefore, fingers are not really free to move too much from the playing position. Among all possible fingers movements that could be exploited, the author opted for a minimal design which focused only on two new gestures of the thumbs of both hands. These consisted in pressing a rather small area adjacent to the finger-holes played by the thumb on ritta and bordone pipes (see Figure 3) that could be exploited without compromising the natural act of playing. This area of each pipe was the easiest to reach by the thumb at any moment of playing. The player could still use the thumb to play the associated notes as usual and was given the possibility to exert and additional pressure when wanted in order to act on a sensor placed therein. Even in presence of open holes during a musical sentence, such an area could still be easily accessed.

In a different vein, the zampogna is an instrument that affords to be moved in various directions without compromising the natural act of playing. Therefore, a set of gestures associated to the orientation of the instrument was defined. Specifically, front-back and left-right movements of the pipes, as well as their combination, were selected because they were the most natural and easiest to perform. The range of each of these movements was defined as not too wide to avoid to hinder the normal playing technique.

3.2 Hardware identification and placement

The hardware technology involved in the augmentation was designed to consist of microphones to capture the contribution of each pipe, sensors used to track the set of new gestures, and a microcontroller board for the digital conversion of the sensors analog values. The overall setup consisted of a soundcard for the digitalization of the microphones signals, a laptop for the processing of such signals and those of the sensors, and a system of loudspeakers for the sound diffusion.

A fundamental design choice was that of tracking separately the sound of each pipe. This was found to be achievable by placing a small microphone in each of the pipes bells. In particular, only the ritta, manca, and bordone pipes were chosen to be enhanced with such microphones. The reason to exclude the fischietto pipe from such an augmentation was due to its cap mechanism, which made not possible the placement of a microphone inside it.

On the one hand, the microcontroller board was designed to be as small as possible in order to be placed easily on the instrument. On the other hand, it was designed to have wireless connectivity in order to avoid the use of a cable connecting it to the external computation unit. Its best placement was identified on the wooden block where the pipes are planted, being this a part of the instrument not touched by the fingers during the act of playing, and where all cables from sensors could most easily merge.

Regarding the technology to sense the identified new gestures, two types of sensors could be involved: pressure sensors, to track the pressure exerted by the thumbs on the areas specified in Section 3.1, and an inertial measurement unit (IMU) to track the position of the instrument. The placement of the latter was identified on the wooden block as that position did not constitute any hinderance to the fingers movements (see Figure 4).

3.3 Mapping strategies

A set of mapping strategies between the player's gestures and the sound production was investigated. It was important to define mappings that were intuitive to the performer and that took into account electronic, acoustic, ergonomic and cognitive limitations. In order to decide on a particular setup, many questions needed to be answered, such as for instance how many parameters of a sound effect the performer could be able to simultaneously control, or how long a performer would need to practice to become comfortable with a particular setup. These mappings were carefully designed to allow a good integration of both acoustic and electronic components of the performance, resulting in an electronically-augmented acoustic instrument that is respectful of the zampogna tradition.

The design of the interactions described in Section 3.1 allowed for the independent modulation of the sound of each

pipe tracked separately. Therefore, one of the mappings that received most attention consisted in the association of a tracked gesture to the sound modulation of a pipe. These associations were defined as:

- the pressure sensor placed on the ritta was associated to the control of the ritta sound, as well as the right-back movement;
- the left-back and left-front movement was associated to the control of the manca sound;
- the pressure sensor placed on the bordone was associated to the control of the bordone sound, as well as the right-front movement.

Another most used mapping focused on the control of both manca and bordone with the same gestures:

- the pressure sensor placed on the ritta was associated to the control of the ritta sound, as well as the right-back movement;
- the pressure sensor placed on the bordone was associated to the control of both manca and bordone sound, as well as the left-back movement.

4. IMPLEMENTATION

4.1 Hardware

The designed augmentation was achieved at hardware level by involving three high quality small microphones, model Sennheiser MKE 1 miniature clip microphone [5], two pressure sensors FSR 400 Force Sensing Resistor [6] manufactured by Interlink Electronics, and the microcontroller board x-OSC [7] manufactured by x-io Technologies Limited.

The x-OSC board was selected for its features: small size, on-board sensors (including an IMU), and wireless transmission of sensors data over Wi-Fi, with a low latency (i.e., 3ms [21]) and via Open Sound Control messages [8]. It was inserted in a plastic box attached with a velcro strip on the wooden block. A velcro strip was also attached to the front part of such plastic box, which allowed the rapid and easy change of attachable batteries.

In order to avoid ruining the wooden parts of the acoustic instrument, a specific low-impact scotch tape strip was placed on all the parts of the instrument where the added hardware was attached. Figure 2 illustrates the Hyper-Zampogna resulting from the augmentation of the zampogna a chiave shown in Figure 1. Figures 3 and 4 illustrate the position of the sensors and microcontroller board in the developed instrument.

[5] http://en-us.sennheiser.com/miniature-clip-on-lavalier-microphone-musicals-live-shows-broadcast-mke-1

[6] http://www.interlinkelectronics.com/FSR400.php

[7] http://www.x-io.co.uk/products/x-osc/

[8] http://www.opensoundcontrol.org/

Figure 2. The developed Hyper-Zampogna.

4.2 Software

A software application was coded in Max/MSP [9] sound synthesis and multimedia platform to implement various sound effects as well as gestures-to-sound parameters mappings. This was achieved by analyzing and processing both the sounds detected from the microphones embedded in the instrument and the data gathered from the sensors.

The placement of the microphones inside the pipes bells allowed for an accurate and separate tracking of ritta, manca and bordone without the need of the application of any further digital signal processing technique to achieve such purpose: the sound of the other pipes was not detected.

The captured acoustic waveforms of each pipe were then processed separately and modulated by the player's interaction with the sensors. This processing consisted in the application of various effects and spatialization techniques. In more detail, various custom effects were implemented mainly involving pitch shifters, vibrato, tremolo, phaser, chorus, wah-wah, parametric equalizers, and dynamics control. Sound spatialization was achieved by defining algorithms used to spatialize virtual sound sources along bi-dimensional and tri-dimensional trajectories in presence of multichannel surround sound systems. For this purpose, the facilities offered by the "Ambisonic Tools for Max/MSP" [22] were exploited.

Moreover, synthesized sound were generated. This was achieved by means of a real-time low latency pitch tracker, whose produced tracked frequencies were utilized to con-

[9] http://www.cycling74.com/

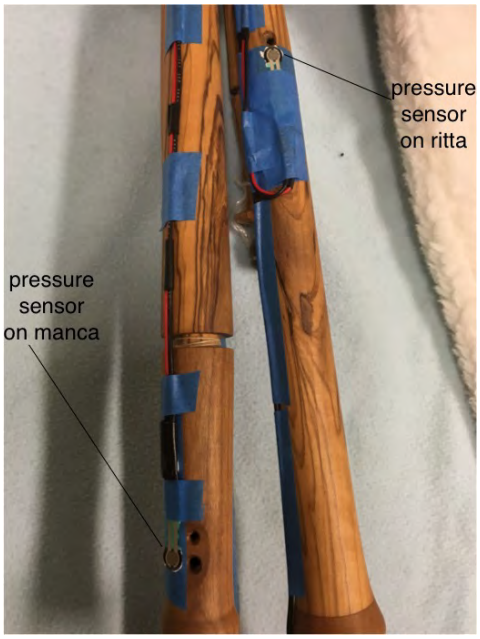


Figure 3. A detail of the involved pressure sensors and their position.

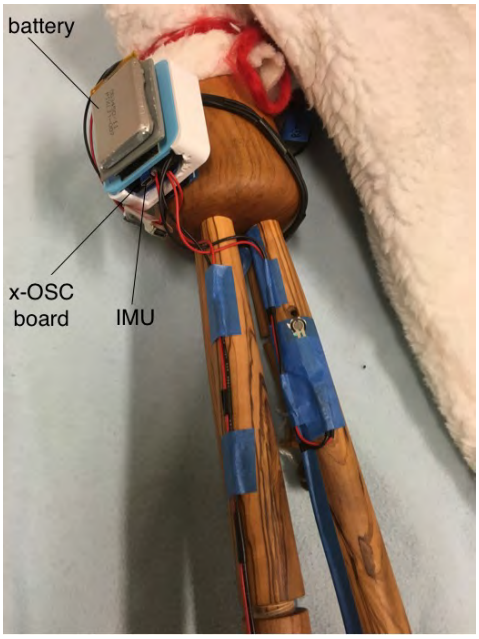


Figure 4. The placement of the wireless microcontroller board with embedded IMU onto the instrument.

trol a custom synthesis module that well merged with the zampogna acoustic sound. The captured sounds before being fed into the pitch tracker underwent a highpass filtering that allowed to achieve an optimal tracking.

Finally, additional mappings were implemented to control various sound effects, synthesizers, loops, and virtual instruments available on the Logic Pro X [10] and Ableton Live [11] digital audio workstations. For this purpose, Max/MSP applications as well as Max for Live devices were implemented, in which the sensors data where processed and converted into MIDI messages.

5. CONCLUSION AND FUTURE WORK

While previous applications of sensors technologies to bagpipes in both academic and industry communities focused on the creation of bagpipe-like controllers for wave-table synthesis or MIDI instruments, this research aimed to an augmentation of the zampogna bagpipe that could fully preserve its original acoustic beheaviour and playing technique. This was achieved by tracking separately the acoustic waveforms of the different pipes as well as modulating both the captured and additional synthesized sounds by means of sensors conveniently located and mapped to digital effects parameters.

The development of the Hyper-Zampogna offered both technical and artistic challenges that the author enjoyed embracing. Analogously to the augmentation of the hurdy-gurdy that he proposed in [11], the augmentation of the zampogna originated from his passion and interest in music technology and traditional instruments, and represents his challenge of combining these two far worlds.

On the one hand, the rationale behind the development of such augmented instrument was to provide electro-acoustic zampogna performers with an interface capable of achieving novel types of musical expression without disrupting the natural interaction with the traditional instrument. On the other hand, this research aimed to provide composers with a new instrument enabling the exploration of novel pathways for musical creation.

The Hyper-Zampogna is currently in a prototype stage and has not been evaluated yet by a zampogna player different from the author. Such an evaluation is planned as well as the use of the Hyper-Zampogna on stage with compositions written by the author.

In future works, the author envisions the extension of the results of this project by means of the creation of a larger palette of sound effects and mapping strategies to control them with the available sensors. In addition, the author plans to augment other types of zampogna different from the zampogna a chiave here involved, such as the "surdulina" and "zampogna gigante" models [19]. A collaboration with a zampogna maker would be beneficial in order to craft from scratch zampogna bagpipes with the sensors and microphones embedded in it.

Finally, it is the author's hope that the results of this research could inspire other builders of augmented instruments to focus on the augmentation of the zampogna bagpipes as well as that composers start writing pieces for it. More information about the Hyper-Zampogna can be found at the author's personal website [12] .

Acknowledgments

This work is part of the "Augmentation of traditional Italian instruments" project, which is supported by Fondazione C.M. Lerici.

[10] http://www.apple.com/logic-pro/
[11] http://www.ableton.com/
[12] http://www.lucaturchet.it

6. REFERENCES

[1] T. Machover and J. Chung, "Hyperinstruments: Musically intelligent and interactive performance and creativity systems," in *Proceedings of the International Computer Music Conference*, 1989.

[2] E. R. Miranda and M. M. Wanderley, *New digital musical instruments: control and interaction beyond the keyboard*. AR Editions, Inc., 2006, vol. 21.

[3] P. R. Cook, "Principles for Designing Computer Music Controllers," in *Proceedings of the International Conference on New Interfaces for Musical Expression*, 2001.

[4] R. Hoadley, "Form and Function: Examples of Music Interface Design," in *Proc. of the HCI2010 Conference*, 2010.

[5] H. Livingston, "Paradigms for the new string instrument: digital and materials technology," *Organized Sound*, vol. 5, no. 3, 2000.

[6] A. Hunt, M. Wanderley, and M. Paradis, "The Importance of Parameter Mapping in Electronic Instrument Design," in *Proceedings of the International Conference on New Interfaces for Musical Expression*, 2002.

[7] D. Arb, J. Couturier, L. Kessous, and V. Verfaille, "Strategies of mapping between gesture data and synthesis model parameters using perceptual spaces," *Organized Sound*, vol. 7, no. 2, 2002.

[8] M. Marshall, M. Hartshorn, M. Wanderley, and D. Levitin, "Sensor choice for parameter modulations in digital musical instruments: Empirical evidence from pitch modulation," *Journal of New Music Research*, vol. 38, no. 3, pp. 241–253, 2009.

[9] A. Kapur, A. J. Lazier, P. Davidson, R. S. Wilson, and P. R. Cook, "The electronic sitar controller," in *Proceedings of the International Conference on New Interfaces for Musical Expression*, 2004, pp. 7–12.

[10] D. Young and G. Essl, "Hyperpuja: A tibetan singing bowl controller," in *Proceedings of the International Conference on New Interfaces for Musical Expression*, 2003, pp. 9–14.

[11] L. Turchet, "The Hyper-Hurdy-Gurdy," in *Proceedings of the Sound and Music Computing Conference*, 2016.

[12] P. Wheeler, "An introduction to bagpipes of the world." *The Journal of the Acoustical Society of America*, vol. 125, no. 4, pp. 2652–2652, 2009.

[13] C. Cannon, S. Hughes, and S. Ó. Modhráin, "Epipe: exploration of the uilleann pipes as a potential controller for computer-based music," in *Proceedings of the international conference on New Interfaces for Musical Expression*, 2003, pp. 3–8.

[14] S. Hughes, C. Cannon, and S. Ó. Modhráin, "Epipe: A novel electronic woodwind controller," in *Proceedings of the international conference on New Interfaces for Musical Expression*, 2004, pp. 199–200.

[15] T. Kirk and C. Leider, "The frankenpipe: a novel bagpipe controller," in *Proceedings of the international conference on New Interfaces for Musical Expression*, 2007, pp. 301–304.

[16] D. Menzies and A. McPherson, "An electronic bagpipe chanter for automatic recognition of highland piping ornamentation." in *Proceedings of the international conference on New Interfaces for Musical Expression*, 2012.

[17] ——, "A digital bagpipe chanter system to assist in one-to-one piping tuition," in *Proceedings of the 2013 Stockholm Music Acoustics Conference/Sound and Music Computing Conference (SMAC/SMC), Stockholm, Sweden*, 2013.

[18] ——, "Highland piping ornament recognition using dynamic time warping," in *Proceedings of the international conference on New Interfaces for Musical Expression*, 2015, pp. 50–53.

[19] M. Gioielli, *La zampogna. Gli aerofoni a sacco in Italia*. Iannone, 2005.

[20] C. Massarelli and A. Valle, "e-zampognë a southern-italian bagpipe controller," in *Proceedings of XVIII Colloquio di Informatica Musicale*, 2010, pp. 75–80.

[21] S. Madgwick and T. Mitchell, "x-osc: A versatile wireless i/o device for creative/music applications," in *Proceedings of Sound and Music Computing Conference*, 2013.

[22] J. Schacher and M. Neukom, "Ambisonics spatialization tools for max/msp," in *Proceedings of the International Computer Music Conference*, 2006.

THE HYPER-HURDY-GURDY

Luca Turchet
Department of Automatic Control
KTH Royal Institute of Technology
turchet@kth.se

ABSTRACT

This paper describes the concept, design, implementation, and evaluation of the Hyper-Hurdy-Gurdy, which is the augmentation of the conventional hurdy-gurdy musical instrument. The augmentation consists of the enhancement of the instrument with different types of sensors and microphones, as well as of novel types of real-time control of digital effects during the performer's act of playing. The placing of the added technology is not a hindrance to the acoustic use of the instrument and is conveniently located. Audio and sensors data processing is accomplished by an application coded in Max/MSP and running on an external computer. Such an application also allows the use of the instrument as a controller for digital audio workstations. On the one hand, the rationale behind the development of the instrument was to provide electro-acoustic hurdy-gurdy performers with an interface able to achieve radically novel types of musical expression without disrupting the natural interaction with the traditional instrument. On the other hand, this research aimed to enable composers with a new instrument capable of allowing them to explore novel pathways for musical creation.

1. INTRODUCTION

During last years numerous exemplars of the so-called "hyper instruments" or "augmented instruments" have been developed [1, 2]. These are conventional acoustic instruments enhanced with sensor and/or actuator technology, and digital signal processing techniques, which serve the purpose of extending the sonic capabilities offered by the instrument in its original version. The performer acting on the sensors can control the production of the electronically generated sounds that complement, or modulate, the sounds acoustically generated by the instrument. The attention of builders of such instruments has focused on the augmentation of various types of acoustic instruments (e.g., violin [3–5], cello [6], saxophone [7], flute [8], trumpet [9], or guitar [10–12] and piano [13]), including the traditional ones (e.g., the uilleann pipes [14], the sitar [15], the Tibetan singing bowl [16], the zampogna [17], or the great highland bagpipe [18]).

This paper presents the augmentation of an instrument typical of the musical tradition of many European countries: the hurdy-gurdy. To the author's best knowledge, prior to this work such a challenge was not faced yet. The hurdy-gurdy is one of the few instruments that can boast not only centuries of history, but also a tradition uninterrupted from Middle Age. When the hurdy-gurdy was born presumably in the Middle Age (its ancestor was called "organistrum") it was certainly one of the musical instruments most advanced of that poque from the technological standpoint. During the course of the history the instrument was subjected to several technical improvements [19]. In particular, in the last decades, innovative instrument makers have made many improvements and additions to the instrument in response to the needs of the hurdy-gurdy players wishing to overcome the technical limitations of the traditional instrument and to extend its sonic possibilities. More strings as well as systems to easily change their intonation were added, so the performer could play in a wider number of tonalities compared to that offered by the traditional version of the instrument. Furthermore, the instrument was enhanced with microphones and entered in the realm of the electro-acoustic instruments.

The author's artistic reflection on the development of an augmented hurdy-gurdy started from these considerations on the history of the instrument and aimed at continuing such a developmental path. The main objective of this research project was to provide the hurdy-gurdy with additional possibilities to allow novel musical expressions, while at the same time avoiding the disruption of the natural interaction occurring between the player and the instrument.

In Section 2 a brief description of the hurdy-gurdy is provided to render this paper more intelligible to those unfamiliar with the instrument.

2. HURDY-GURDY DESCRIPTION

The hurdy-gurdy (see Figure 1) is a stringed musical instrument whose sound is produced by turning a crank that controls a wheel rubbing against the strings. Such a wheel is covered with rosin and functions much like a continuous violin bow. The vibration of the strings is made audible thanks to a soundboard. Melodies are played on a keyboard that presses small wedges against one or more strings (called "chanterelles") to change their pitch. Moreover, hurdy-gurdies have multiple "drone string", which provide a constant pitch accompaniment to the melody. Each of the strings can be easily put on or removed from the contact with the wheel.

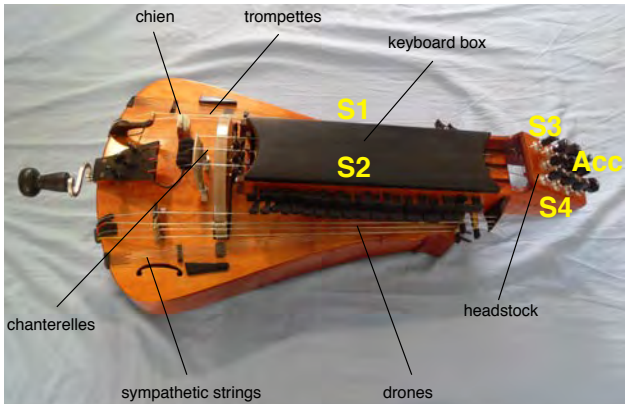


Figure 1. An exemplar of electro-acoustic hurdy-gurdy with the indications of its main components and the identified sensors positions.

Hurdy-gurdies are able to provide percussive sounds produced by means of one or more buzzing bridges. These are called "chiens", act like a sort of hammer having a tail and a free end, and are placed under one or more drone strings called "trompettes". The tail of such chiens is inserted into a narrow vertical slot that holds them in place, while their free end rest on the soundboard and is more or less free to vibrate. It is precisely the vibration of the free ends of the chiens that produce the unmistakable percussive sound of the instrument: when the wheel is turned slowly the pressure on the trompettes strings holds the chien in place, sounding a drone, while when the crank is accelerated, the hammer lifts up and vibrates against the soundboard producing the characteristic buzzing noise. Such a buzz is used as an articulation or to provide rhythmic percussive effects.

Recently a new model of electro-acoustic hurdy-gurdy has been crafted by the luthier Wolfgang Weichselbaumer [1] (see Figure 1). One of the many novelties lies in the complex system of six embedded microphones placed in as many parts of the instrument in order to track the sound of each component: one piezo-electric microphone is placed under the buzzing noise bridge capable of detecting mainly the contribution to the instrument sound given by the trompettes strings; one piezo-electric microphone is placed under the wooden part where the drones were positioned, capable of tracking mainly their contribution; one piezo-electric microphone is placed under the bridge of the chanterelles positioned capable of tracking mainly their contribution; two one piezo-electric microphones are placed in correspondence of the two sets of sympathetic strings, capable of tracking mainly their contribution; one omni-directional small microphone placed near the chanterelles bridge, capable of tracking the overall acoustic sound of the instrument. Each of the five present microphones is able to track with high precision the richness of the sound of each component. Such a microphone system is at the basis of the augmentation presented in this paper.

[1] http://www.weichselbaumer.cc/

3. MAIN CONCEPTS

The first step to satisfy the goal of augmenting the hurdy-gurdy to achieve a novel interface for musical expression, capable to open radically new paths for composition and performance, consisted in determining the needs and conditions to meet for the new instrument. This research started by the author's questioning about his personal needs, as a performer, of extending the sonic possibilities of the instrument and overcoming its limitations when used in conjunction with the most widespread current technologies for sound processing. Such need resulted in the following requirements.

The first requirement consisted of enhancing the instrument without physically modifying it with holes, carvings or attaching new pieces of wood for instance: the technology should have been easy to put on and remove, and the instrument could have been still played in the normal acoustic way, if wanted.

The second requirement was to augment the instrument in such a way that the conventional set of gestures to play the instrument would remain unaltered: the instrument should have kept working in the conventional way after the augmentation. For this purpose, the way of playing the instrument was analyzed in order to identify the possible set of new gestures that a performer would act on the instrument without interfering with the natural act of playing. The right hand appeared immediately the most difficult to act on. This was due to the complexity of tracking the quick and subtle movements (especially small variations in acceleration) of the wheel, wrist and fingers while turning the crank. A solution was attempted by placing some accelerometers attached to the wrist, but the tracking resulted not to be optimal due to accuracy and latency issues. A possible solution to track the wheel would have been that of using magnets inserted into it and leveraging the so-called "hall effect". However, these solutions would have required the performer to wear some sensors (e.g., wireless bracelets, or wireless boards with embedded accelerometers), which would have been perceived as obtrusive, or to groove some carvings into the wheel to put magnets and cope with the problem of having some cumbersome cables placed on the instrument: this not only would have limited the ease of playing and even of moving the instrument, but also would have affected the robustness of the added technology. For these reasons, the research was focused on the tracking of the left hand gestures and of the orientation of the instrument.

The third requirement consisted of limiting as much as possible the unwanted interactions of the performer with the technology added to the instrument different from the sensors. This resulted in reducing at the minimum the amount and the length of the involved wires, and to hide as much as possible the technology inside the instrument as well as by adopting wireless solutions.

The fourth requirement was to allow hurdy-gurdy performers to achieve unprecedented sound modulations. In first place, this consisted of enabling the possibility of exerting a strict control of a sound effect at note level. Indeed, by means of current technologies a hurdy-gurdy performer

can use an effect (e.g., a delay) to control the sound modulation of a whole musical sentence, but can not apply that particular effect on a single note of the musical sentence and keep the other notes unaffected by that effect. In second place, the augmentation had to provide the possibility to modulate separately the sound produced by the various components of the instrument (see Section 2). This could be only possible by involving a set of microphones and a palette of signal processing algorithms capable of detecting and isolating such components. In third place, performers had to be able to avail themselves of sound effects specifically built for the various components of the instrument that could allow to transcend the physical limitations of the instrument itself. For instance, smooth and long glissando and bending are not possible on the traditional instrument. Analogously, the frequency of a drone could be modulated to add some vibrato (thing not possible on the conventional instrument since the drones are not pressed by the fingers) or the sound of a single chanterelle could be transformed into a bi-chord.

4. DESIGN

4.1 New gestures identification

The design process started with the identification of a new possible set of gestures that could be reasonably added to the normal playing technique without disrupting it. The most important of these are the following:

- while playing the chanterelles by means of the fingers acting on the keys of the keyboard, the thumb is normally free and can be exploited to press an area of the keyboard or slide upon it;
- the pinkie can be used to press a key, the index to press an area of the keyboard, and the thumb to press another another area of the instrument placed at a even larger distances from the keyboard;
- when the fingers are not involved in acting on the keys (e.g., when chanterelles are used to produce their sound as open strings, or when sympathetic strings are plucked) the left hand is totally free and different fingers could press/slide on various areas of the instrument even very far from the keyboard;
- all these new added gestures, as well as the ones of the conventional playing technique, can be performed simultaneously with tilting up and down or forth and back the whole instrument.

4.2 Hardware technology identification an placement

The technology involved in the augmentation (additional to the set of embedded microphones already present) was designed to consist of sensors used to track the set of new gestures and a microcontroller board for the digital conversion of the sensors analog values. Three types of sensors could be involved:

- pressure sensors, to track pressure of the fingers on an area of the instrument;
- ribbon sensors, to track the position of the fingers on an area of the instrument;
- accelerometers to track the tilting of the instrument.

A first design choice was that pressure and ribbon sensors had to cover relatively wide areas in order to achieve an optimal accessibility. The use of strip-shaped sensors of various lengths was considered the optimal choice for this purpose. A second design choice was to place a ribbon sensor on top of a pressure sensor in order to detect simultaneously the information about the pressure force exerted by the finger as well as its position on a certain part of the instrument. The microcontroller board was designed to be as small as possible in order to be placed easily on the instrument, and to have wireless connectivity in order to avoid the use of a cable connecting it to an external computation unit responsible for processing both the microphones and the sensors signals.

The number and placement of the identified sensors and microcontroller board represented a challenging problem due to the complexity of the shape of the hurdy-gurdy, the hardware limitations of the sensors themselves, and the set requirement of keeping unaltered the natural interaction of the player with the instrument. Four pairs of pressure-ribbon sensors and one 3-axis accelerometer were chosen. The four pairs of sensors were placed on top of the keyboard box (see "S1" in Fig. 1); at the side of the keyboard box (see "S2" in Fig. 1); on the top of the headstock (see "S3" in Fig. 1); on the bottom of the headstock (see "S4" in Fig. 1). These positions were chosen for their easiness in reachability with the fingers and because they did not interfere neither with the normal way of playing nor with the functioning of the various components of the instrument. The best position to place the accelerometers was identified to be on the interior part of the headstock (see "Acc" in Fig. 1), since it did not interfere with the placement of the other sensors and could easily be attached to the instrument. The best position for the microcontroller board was also identified as the space behind the headstock. This choice was motivated by the fact that the wires coming out from the sensors could reach the board easily, with the shortest distance, and without interfering with the functioning of the various components of the instrument. In addition, in that position the board was hidden from the sight and above all it could be naturally protected from unwanted collisions.

4.3 Mapping strategies

The design for the interactive control of the developed instrument was based both on the extraction of features from the data captured by sensors and from the acoustic waveforms captured by microphones. A set of mapping strategies between the performers gestures and the sound production was investigated. It was important to define mappings that were intuitive to the performer and that took into account electronic, acoustic, ergonomic and cognitive limitations. In order to decide on a particular setup, many questions needed to be answered, such as for instance how many parameters of a sound effect the performer could be

able to simultaneously control, or how long a performer would need to practice to become comfortable with a particular setup.

The hurdy-gurdy is an instrument with an intrinsic high level of affordances as far as the features suitable for the control of the digital sound production are concerned. It can be used as a percussive, melodic and accompanying instrument, and from all of these characteristics it is possible to find a variety of potential controls by extracting acoustic features from the sound captured by the microphones. These controls can be used in conjunction with those resulting from the interaction with sensors.

The first step in the mappings design process consisted of defining associations of each pair of sensors to a component of the instrument. Sensors placed in positions S1, S2, S3, and S4 indicated in Fig. 1 were mainly used to control the sound captured by the microphones of the chanterelles, trompettes, sympathetic strings, and drones respectively. Nevertheless, such associations could change in such a way that the same pair of sensors could control more than one instrument component, or, vice versa, more than one pair of sensors could control a single instruments component. The second step consisted of the definition of the mappings between the performers gestures acted on the sensors and the parameters of the selected algorithms for the various sound effects. These mappings were carefully designed to allow a good integration of both acoustic and electronic components of the performance, resulting in an electronically-augmented acoustic instrument that is respectful of the hurdy-gurdy tradition.

5. IMPLEMENTATION

5.1 Hardware

The designed augmentation was achieved at hardware level by involving the pressure sensors FSR 408 Strip Force Sensing Resistor [2] manufactured by Interlink Electronics, the ribbon sensors Soft Pot [3] manufactured by Spectra Symbol, and the microcontroller board x-OSC [4] manufactured by x-io Technologies Limited.

In each of the four ribbon-pressure sensor pairs, the ribbon sensor was attached, thanks to its adhesive film, on top of the pressure sensor in order to create a unique device capable of providing simultaneous information about position and pressure of the finger interacting with it. The pressure sensor was in turn attached, thanks to its adhesive film, to a plastic rigid support, which was appropriately cut in order to meet the size of the sensors. This support was involved for two reasons. The first one was that placing the sensors directly on the instrument did not allow an optimal tracking of the forces and positions exerted by the fingers on the sensors due to the fact that in some cases (e.g., the keyboard box) the wood could slightly move up and down, and a more rigid, homogenous, and stable base was needed. The second one was that thanks to the support the created device could be easily attached or removed to the instrument. In order to avoid ruining the wooden parts of the acoustic instrument, a specific low-impact scotch tape strip was placed on the part of the instrument where the plastic support was attached.

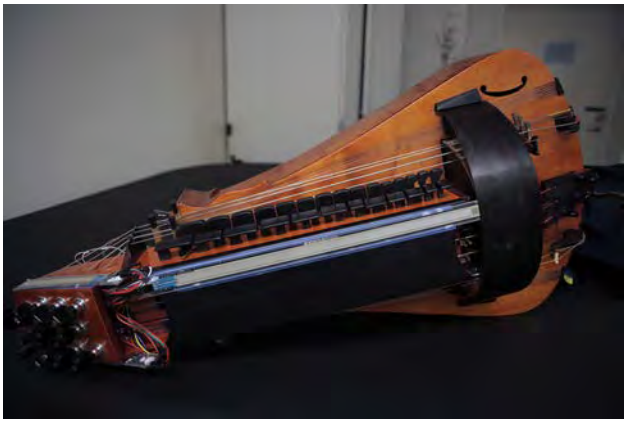


Figure 2. The developed Hyper-Hurdy-Gurdy.

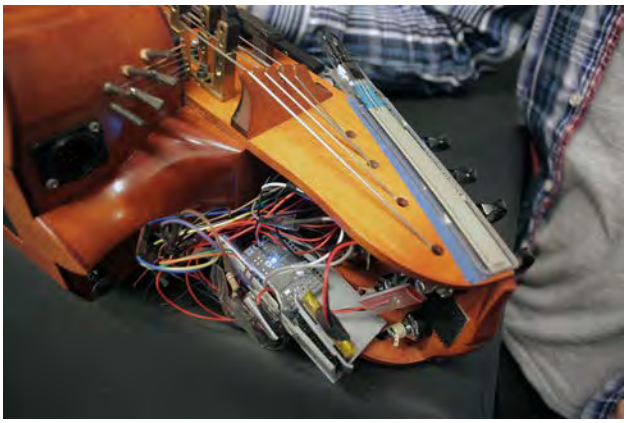


Figure 3. The placement of the wireless microcontroller board on the instrument.

The x-OSC board was selected for its features: small size, on-board sensors (including a 3-axis accelerometers), and wireless transmission of sensors data over WiFi, with a low latency (i.e., 3ms [20]) and via Open Sound Control messages [5]. Figures 2 and 3 illustrate the position of the sensors and microcontroller board in the developed instrument.

5.2 Software

As far as the software is concerned, the Max/MSP [6] sound synthesis and multimedia platform was utilized. An application was coded to implement the designed sound effects and mappings, by analyzing and processing both the sounds detected from the microphones embedded in the instrument and the data gathered from the sensors.

The first issue encountered was that the microphones were not effective in detecting separately each of the components of the instrument. For instance, the sound produced by the drones was in part detected by the microphones of

[2] http://www.interlinkelectronics.com/FSR408.php
[3] http://www.spectrasymbol.com/potentiometer/softpot
[4] http://www.x-io.co.uk/products/x-osc/
[5] http://www.opensoundcontrol.org/
[6] http://www.cycling74.com/

the chanterelles; similarly, the microphone of the trompettes detected also the sound of the chanterelles. A complete isolation of such components is not possible in an acoustic instrument such as the hurdy-gurdy since the vibrations produced by one component propagate everywhere in the instrument and are detected by contact microphones or external microphones placed in a whatever part of the instrument. Therefore, some signal processing techniques were needed to achieve the goal of isolating as much as possible the sound of each component in order to process it separately. For instance, a low pass filter was applied to the input signal coming from the microphone of the drones in order to limit the amount of signal resulting from playing the chanterelles. Vice versa, a high pass filter was applied to the signal coming from the contact microphone placed on the chanterelles bridge to limit both the low frequencies produced by the drones and of the noise of resulting from pressing the keys. Ad hoc signal processing algorithms were also implemented for analyzing the captured acoustic waveforms in order to achieve particular sound effects. For example, to extract only the buzzing noise component from the sound produced by the trompettes, a signal gate was involved which was activated according to a threshold set on the sound amplitude. The specific research challenge in using all the algorithms for processing the captured acoustic waveforms was that of finding the best combination of the algorithms parameters in order to achieve the best result.

Furthermore, in presence of the hits on the crank made in order to produce the buzzing noises, the resulting impulsive variation in the acceleration tracked by the accelerometers needed to be excluded. To solve such issues, various mean filters, median filters, and low pass filters, were applied. These processing techniques were effective in smoothing the rapid variations happening in the signal. However, their application had the side effect of introducing latency. Therefore, a large amount of research consisted in finding the right values for the parameters of such filters in order to achieve the best tradeoff between the accuracy in tracking and the latency of the response produced by the filters.

Once a good tracking of both performer's gestures and instrument components sounds was achieved, several mappings were implemented. Examples of these are the following [7] :

- The amount of volume of a sound effect was mapped to the amount of pressure exerted by a finger on a pressure sensor, such that when the sensor was not pressed the effect was not activated, and when it was pressed the presence of the effect could be modulated individually for each note.
- The sliding of the finger on a ribbon sensor was mapped on the amount of frequency transposition in a pitch shifting algorithm such as the glissando effect could be produced.
- The combination of the use of both the pressure and ribbon sensors for the previous two mappings resulted on a glissando effect whose activation depended on the presence of the finger on the sensor, the frequency transposition depended on the finger position, and the volume depended on the amount exerted pressure force.
- The amount of up-down or back-forth tilting movements tracked by accelerometers was mapped to the activation of an effect: when the amount of tilting overcame a certain threshold the effect was activated. This way of using the tilting as a switch for an effect rather than a continuous control was due to the fact that great displacements from the normal position of the instrument could be tracked in a easier way and were subjected to less variations. Indeed the rapid and strong movements produced while playing the hurdy-gurdy with the buzzing noise of the trompettes could lead to impulsive variations in the signal acquired by the accelerometers, and this could not adapt well for a continuous control usage.

Moreover, a variety of mappings were defined on the basis of algorithms used to spatialize virtual sound sources along bi-dimensional and tri-dimensional trajectories in presence of multichannel surround sound systems. For this purpose, the facilities offered by the "Ambisonic Tools for Max/MSP" [21] were used.

Finally, additional mappings were implemented to control various sound effects, synthesizers, loops, and virtual instruments available on the Logic Pro X [8] and Ableton Live [9] digital audio workstations. For this purpose, Max/MSP applications as well as Max for Live devices were implemented, in which the sensors data where processed and converted into MIDI messages.

6. EVALUATION

The developed instrument was subjected to extensive tests aimed to validate the implemented augmentation from the technological and expressive standpoints. In addition to the author's own evaluation, the Hyper-Hurdy-Gurdy was tested by Johannes Geworkian Hellman [10] , a well known hurdy-gurdy performer and virtuoso. The testing session was conducted in a acoustically isolated room of the KMH Royal College of Music of Stockholm and lasted about one hour. The setup consisted of the developed Hyper-Hurdy-Gurdy configured to have all sensors mapped to at least one parameter of a sound effect, a soundcard (Fireface UFX), two loudspeakers (Genelec 8050B Studio Monitor), and a laptop (Macbook Pro) running the software applications described in Section 5.2.

The session consisted of three parts, which took about 10, 35, and 15 minutes respectively. In the first part the performer was asked to interact with the instrument without receiving any information about the added technology. This procedure was adopted in order to assess the very first

[7] A comprehensive list of audio-visual examples of the implemented mappings is available at:
https://www.youtube.com/watch?v=9c1QFg2bG9w

[8] http://www.apple.com/logic-pro/

[9] http://www.ableton.com/

[10] http://www.johannesgeworkianhellman.com/

approach with the instrument. During this part, only the four pair of sensors were explored. The mappings related to the accelerometers were not detected. This was due to the fact that the position of the instrument was not tilted and the accelerometers, differently from the other sensors, were not visible. The associations between the sensors and the corresponding controlled components of the instrument were all identified and understood.

In the second part the various sensors and mappings were explained, and questions were made regarding the appropriateness of the sensors position, intuitiveness of the mappings involved, and the effectiveness of the types of sound effects utilized. The performer reported to have appreciated the fact that the sensors were placed in ergonomic ways, they were easy to reach while normally playing, and they did not require too much force to be activated. Moreover, very positive comments were reported about the effectiveness of all the implemented mappings, in particular about the appropriateness and accuracy of all the involved ranges of the parameters. One of the most relevant comments was *"With this instrument I can easily apply and control an effect to each note I produce, so now I can do things that I could not achieve with the controls for the effects I normally use."* Interestingly, from some comments it emerged the need of having available some discrete controls in addition to the continuous ones present.

In the third part, the performer was asked to play the instrument, taking advantage of the new affordances offered by the instrument and exploring the novel possibilities for improvisation. As one would expect a final comment was *"I think one would need a lot of exploration and experience to learn how to really use these new possibilities"*. Nevertheless, overall, his feedback was very positive and confirmed the goodness of the author's design choices.

7. HYPER-HURDY-GURDY IN LIVE PERFORMANCE

The Hyper-Hurdy-Gurdy has been used for musical creations and performances purposes. It was premiered at the Audiorama concert venue in Stockholm in April 2015. A 21-channels composition, named "Incantesimo", for solo Hyper-Hurdy-Gurdy was performed. Subsequently, various pieces were composed and performed by the author both as a soloist and in chamber orchestra. Videos documenting a technical demonstration of the Hyper-Hurdy-Gurdy and its usage in live performances are available on the author's personal website [11] . Those live performances constitute the final validation of the developed instrument.

8. CONCLUSIONS AND FUTURE WORK

On the one hand, the rationale behind the development of the instrument was to provide hurdy-gurdy performers with an interface able to achieve novel types for musical expression without disrupting the natural interaction with the traditional instrument. On the other hand, this research aimed to enable composers with a new instrument capable of allowing them to explore novel pathways for musical creation. The proposed research resulted in an augmented instrument suitable for the use in both live performance, improvisation, and composition contexts. Novel timbres and forms of performer-instrument interactions were achieved, which resulted in an enhancement of the conventional electro-acoustic performances as well as in a variety of new compositional possibilities.

This augmentation of the traditional hurdy-gurdy originated from the author's two passions and interests: traditional instruments and music technology. The development of the Hyper-Hurdy-Gurdy and the compositions for it represent the author's challenge of combining these two far worlds. This research was motivated by the author's need to investigate new paths for individual musical expressions as well as to research how to progress the possibilities for music creation with the hurdy-gurdy and electronics normally associated to it. At the conclusion of the project, it is the author's opinion that the developed instrument is effectively capable of responding to such needs. Undoubtedly, these needs are also shared by many musicians and composers who constantly search for novel tools and ideas for their artistic works. However, in the author's vision, completely novel paths are not practically possible with the current conventional acoustic and electro-acoustic hurdy-gurdies, since basically all the expression possibilities available with them have been already investigated. With the introduction of a novel generation of Hyper-Hurdy-Gurdies, the possibilities for absolutely novel musical research paths are countless, and revolutionary approaches to composition and improvisation can be explored. The pieces that the author composed and performed might be considered as a proof of these statements.

As far as future works are concerned, the author envisions various possibilities for extending the results of this project. First of all the collaboration with an instrument maker would be beneficial in order to craft from scratch a hurdy-gurdy with the sensors embedded in it. Secondly, different types as well as a larger number of sensors could be added. In particular a set of small and fully configurable buttons and knobs placed onto the instrument would be useful to change presets of sounds effects and/or mappings: this would allow to avoid the use of external tools dedicated for this purposes such as footpedals. Furthermore, an actuated system could be added in a way similar to that proposed for the actuated violin presented in [22] or the smart guitar developed by Mind Music Labs [23].

Finally, it is the author's hope that the results presented in this paper could inspire other digital luthiers, performers, and composers to continue this research on augmenting the hurdy-gurdy as well as on composing for it.

Acknowledgments

This work is part of the "Augmentation of traditional Italian instruments" project, which is supported by Fondazione C.M. Lerici. The author acknowledge the hurdy-gurdy performer Johannes Geworkian Hellman for having participated to the evaluation of the developed instrument.

[11] www.lucaturchet.it

9. REFERENCES

[1] T. Machover and J. Chung, "Hyperinstruments: Musically intelligent and interactive performance and creativity systems," in *Proceedings of the International Computer Music Conference*, 1989.

[2] E. R. Miranda and M. M. Wanderley, *New digital musical instruments: control and interaction beyond the keyboard*. AR Editions, Inc., 2006, vol. 21.

[3] F. Bevilacqua, N. Rasamimanana, E. Fléty, S. Lemouton, and F. Baschet, "The augmented violin project: research, composition and performance report," in *Proceedings of the International Conference on New Interfaces for Musical Expression*, 2006, pp. 402–406.

[4] D. Overholt, "Violin-related HCI: A taxonomy elicited by the musical interface technology design space," in *Arts and Technology*. Springer, 2012, pp. 80–89.

[5] L. S. Pardue, C. Harte, and A. P. McPherson, "A low-cost real-time tracking system for violin," *Journal of New Music Research*, vol. 44, no. 4, 2015.

[6] A. Freed, D. Wessel, M. Zbyszynsky, and F. Uitti, "Augmenting the cello," in *Proceedings of the International Conference on New Interfaces for Musical Expression*, 2006.

[7] S. Schiesser and C. Traube, "On making and playing an electronically-augmented saxophone," in *Proceedings of the International Conference on New Interfaces for Musical Expression*, 2006.

[8] C. Palacio-Quintin, "Eight Years of Practice on the HyperFlute: Technological and Musical Perspectives," in *Proceedings of the International Conference on New Interfaces for Musical Expression*, 2008.

[9] J. Thibodeau and M. M. Wanderley, "Trumpet augmentation and technological symbiosis," *Computer Music Journal*, vol. 37, no. 3, 2013.

[10] N. Bouillot, M. Wozniewski, Z. Settel, and J. R. Cooperstock, "A mobile wireless augmented guitar." in *Proceedings of the International Conference on New Interfaces for Musical Expression*, 2008, pp. 189–192.

[11] O. Lähdeoja, M. M. Wanderley, and J. Malloch, "Instrument augmentation using ancillary gestures for subtle sonic effects," *Proceedings of the Sound and Music Computing Conference*, pp. 327–330, 2009.

[12] O. Lahdeoja, "An augmented guitar with active acoustics," in *Proceedings of the Sound and Music Computing Conference*, 2015.

[13] A. McPherson, "Buttons, handles, and keys: Advances in continuous-control keyboard instruments," *Computer Music Journal*, vol. 39, no. 2, pp. 28–46, 2015.

[14] C. Cannon, S. Hughes, and S. Ó. Modhráin, "Epipe: exploration of the uilleann pipes as a potential controller for computer-based music," in *Proceedings of the international conference on New Interfaces for Musical Expression*, 2003, pp. 3–8.

[15] A. Kapur, A. J. Lazier, P. Davidson, R. S. Wilson, and P. R. Cook, "The electronic sitar controller," in *Proceedings of the International Conference on New Interfaces for Musical Expression*, 2004, pp. 7–12.

[16] D. Young and G. Essl, "Hyperpuja: A tibetan singing bowl controller," in *Proceedings of the International Conference on New Interfaces for Musical Expression*, 2003, pp. 9–14.

[17] L. Turchet, "The Hyper-Zampogna," in *Proceedings of the Sound and Music Computing Conference*, 2016.

[18] D. Menzies and A. McPherson, "An electronic bagpipe chanter for automatic recognition of highland piping ornamentation." in *Proceedings of the international conference on New Interfaces for Musical Expression*, 2012.

[19] S. Palmer and S. Palmer, *The hurdy-gurdy*. David and Charles, Brunel House Newton Abbot Devon, UK, 1980.

[20] S. Madgwick and T. Mitchell, "x-osc: A versatile wireless i/o device for creative/music applications," in *Proceedings of Sound and Music Computing Conference*, 2013.

[21] J. Schacher and M. Neukom, "Ambisonics spatialization tools for max/msp," in *Proceedings of the International Computer Music Conference*, 2006.

[22] D. Overholt, E. Berdahl, and R. Hamilton, "Advancements in actuated musical instruments," *Organised Sound*, vol. 16, no. 02, pp. 154–165, 2011.

[23] L. Turchet, A. McPherson, and C. Fischione, "Smart instruments: Towards an ecosystem of interoperable devices connecting performers and audiences," in *Proceedings of the Sound and Music Computing Conference*, 2016.

SMART INSTRUMENTS: TOWARDS AN ECOSYSTEM OF INTEROPERABLE DEVICES CONNECTING PERFORMERS AND AUDIENCES

Luca Tuchet
Department of Automatic Control
KTH Royal Institute of Technology,
MIND Music Labs
turchet@kth.se

Andrew McPherson
Centre for Digital Music
School of EECS
Queen Mary University of London
a.mcpherson@qmul.ac.uk

Carlo Fischione
Department of Automatic Control
KTH Royal Institute of Technology,
MIND Music Labs
carlofi@kth.se

ABSTRACT

This paper proposes a new class of augmented musical instruments, "Smart Instruments", which are characterized by embedded computational intelligence, bidirectional wireless connectivity, an embedded sound delivery system, and an onboard system for feedback to the player. Smart Instruments bring together separate strands of augmented instrument, networked music and Internet of Things technology, offering direct point-to-point communication between each other and other portable sensor-enabled devices, without need for a central mediator such as a laptop. This technological infrastructure enables an ecosystem of interoperable devices connecting performers as well as performers and audiences, which can support new performer-performer and audience-performer interactions. As an example of the Smart Instruments concept, this paper presents the Sensus Smart Guitar, a guitar augmented with sensors, onboard processing and wireless communication.

1. INTRODUCTION

Digital musical instrument design often involves a balance of seeking artistic and technical novelty while connecting with established musical traditions and playing techniques. Augmented instruments [1, 2] have a long history of extending the creative possibilities of familiar acoustic instruments through sensors, actuators and signal processing techniques.

Communication is fundamental to any musical performance, whether it is amongst performers or between performers and audience. Networked musical performance has an established history [3], but further opportunities exist for networking amongst augmented instruments, many of which are bespoke self-contained systems.

A useful model for interconnected musical instruments comes from the *Internet of Things* (IoT), an umbrella term encompassing the augmentation and interconnection of physical devices [4]. Recent years have seen a substantial expansion in "smart" devices and appliances in the home, office and other environments which connect wirelessly through the internet to other more conventional computing devices. However, IoT integration in musical instruments has thus far received comparatively little attention.

In this paper we propose a novel class of musical instruments, *Smart Instruments*, which are characterized by embedded intelligence, bidirectional wireless connectivity and an embedded sound delivery system. [1] Smart Instruments integrate disparate technologies found in various strands of augmented instrument, networked music and IoT research. They offer a direct point-to-point communication between each other and other portable sensor-enabled devices, without need for a central mediator such as a laptop. In this paper, we suggest that the holistic Smart Instruments approach will enable artistic capabilities beyond current augmented instruments.

Section 2 of this paper examines the component parts of the Smart Instruments concept, including recent developments in sensor-augmented and actuated musical instruments, lutherie techniques, and relevant IoT technologies. Section 3 then argues for the prospect of a holistic integration of these technologies in a new generation of Smart Instruments. An example of this approach, the Sensus Smart Guitar, is presented in Section 4, and future prospects are discussed in Section 5.

2. RELATED WORK

2.1 Augmented Instruments

The augmentation of familiar acoustic instruments with sensor technologies has a long history [2,7]. More recently, interest has grown in electromechanically actuating the vibrating structures of acoustic instruments [8]. Though a comprehensive survey of these efforts is beyond the scope of this paper, this section examines selected recent examples in this space.

2.1.1 Sensor Strategies

The addition of sensors to familiar instruments is well-established, as is the construction of "instrument-like controllers" [9], which replicate the physical form of a famil-

[1] Our use of the term "Smart Instruments" is distinct from the IRCAM *SmartInstruments* active acoustics project (e.g. [5, 6]), though onboard acoustic actuation is one component of a Smart Instrument in our usage. Full details on the IRCAM SmartInstruments can be found at http://instrum.ircam.fr/smartinstruments/

iar instrument to control other sounds. Sensor augmentations exist of nearly every familiar instrument, including violin [10–12], trumpet [13], guitar [14–16] and piano [17], as well as more rare instruments, such as the hurdy-gurdy [18]. Techniques have been proposed for customisable sensor surfaces adaptable to different applications [19] and toolkits for musicians to create their own augmentations [20].

In some cases, sensors are used as extra controls separate from the main playing techniques, while other approaches seek to provide a more detailed picture of playing gesture than audio alone can provide [10, 12]. Other sensor approaches use ancillary gestures to control sonic effects [15] without requiring the performer to explicitly manipulate additional controls.

In many cases, the addition of sensors requires either wired connections to outboard computing, though wireless communication links [14], embedded computing [21] and fixed camera-based sensing [22] are also used to give the performer free movement.

2.1.2 Actuated Acoustic Instruments

Sensor augmentation of instruments often relies on audio post-processing or digital sound synthesis played through external loudspeakers. More recent developments fold the sound synthesis back into the acoustic structure of the instrument. These *actuated instruments* [8] seek to retain the sonic richness of acoustic instruments while expanding their performance possibilities.

Actuation can be applied directly to an instruments vibrating elements: examples include guitar strings [23], piano strings [24,25], drum heads [26], vibraphone bars [27] and metal tines in a Fender Rhodes [28]. Feedback control allows the application of novel effects including active damping [23], changing resonance properties [29], inducing self-sustaining oscillations [5] and creating novel timbre effects [6]. In other cases, speakers or vibration actuators are embedded in the resonant chamber of an instrument to manipulate its sound [16, 30, 31].

2.2 Digital and Hybrid Lutherie

Lutherie, the technique of building musical instruments, requires artistry, skilled craftsmanship and intimate knowledge of the materials one works with. This is no less true in *digital lutherie* [9] than the acoustic techniques that preceded it. Considerations in the digital domain relate not only to sensors and synthesis techniques but to the mapping strategies between them, a review of which is beyond the scope of this paper.

An emerging practical consideration within digital lutherie is creating entirely self-contained instruments using embedded single-board computers [32]. While self-contained digital instruments once required specialist DSP platforms and significant engineering resources, the rise of embedded computers like Raspberry Pi and BeagleBone Black and associated audio maker platforms [21,33] has increased the accessibility of self-contained instruments.

Digital lutherie does not imply an inattention to physical materials. While 3D printing and other rapid prototyping technologies have been applied to create acoustic instruments [34], materials and physical craft often feature prominently in new hybrid acoustic-electronic instruments such as the Halldorophone by Halldor Ulfarsson [2] and the Overtone Fiddle [35].

2.3 Wireless Sensor Networks

Wireless sensors networks (WSNs) [36] are the essential component of IoT. They are networks of tiny autonomous sensor and actuator nodes that can be embedded in any physical object for control and monitoring via wireless transmission. They are characterized by the scarcity of resources for communication, computation and energy supply. However, despite WSNs are making smarter many devices such as phones, watches, and home electronic appliances, little has been done when it comes to musical instruments manufacturing. The capability to deploy small sensing nodes everywhere, without the need of power supply and cables, makes these networks a most interesting and versatile technology to embed in the musical instruments in a seamless manner and without impeding the traditional interaction with the instruments.

There have been many efforts to design WSNs, both in academia and industry (e.g., [37]). New communication protocols for WSNs have been built around standardized low-power protocols such as IEEE [38], Zigbee [3] , ROLL [4] . The design of WSNs for musical instruments should be grounded in the theory of cross layer design [39] to overcome problems such as message losses, delays and lack of synchronization among sensor nodes. This poses difficult design challenges, especially in the musical domain, where the transmission of messages has to be very reliable and the communication latencies very short.

3. THE SMART INSTRUMENTS

From the analysis of the works reviewed in previous section, it emerges that various systems have been developed to satisfy different needs, but that such systems have not been integrated yet. Musical instruments augmented with sensors for gesture tracking respond to the need of controlling the sound output in novel ways in order to achieve novel types of musical expressions. Instruments augmented with actuation systems satisfy the need of modulating the vibrations of the resonant body. Actuated systems, as well as loudspeakers systems embedded in the instrument, serve the purpose of having the source of the electronically generated sounds placed onto the instruments. IoT technologies satisfy the need of having a bidirectional communication between two or more devices via wireless connectivity. Embedded systems serve the purpose of having the computational unit placed inside the musical instrument. Systems for collaborative networked music respond to the need of exploring novel forms of music creation. Furthermore, current digital audio workstations (DAWs) serve the

[2] https://www.youtube.com/watch?v=uo4Jq-_tysc
[3] http://www.zigbee.org
[4] http://www.ietf.org/dyn/wg/charter/roll-charter.html

purposes of applying effects to audio signals, generate synthesized sounds, as well as mix, record and play audio tracks.

We argue that the integration of all the technologies described in those works will lead to a novel class of musical instruments that we define as "Smart Instruments".

3.1 Features

The class of Smart Instruments that we propose are characterized by the following components:

- a system for capturing the sonic output generated by the instrument (e.g., by means of microphones or pickups embedded in the instrument);
- a system to extract in real-time the musical information related to the player's interaction (e.g., by means of pitch tracker, onset detection, and envelope following algorithms);
- a system for networked, bidirectional, low-latency, and wireless communication of various kinds of data (including audio streams) towards/from connected devices (including other smart instruments). Such a system can leverage both the Internet and ad-hoc communication networks;
- a sensors-based system for tracking a performer's gestures. The tracked interactions with the sensors are used to modulate the instrument sonic output and to deliver control messages to connected external devices;
- a sound delivery system located onto the instrument (for instance via actuation systems or loudspeakers embedded in the instrument);
- an embedded computational unit for sensors data processing, for sound processing and generation including all capabilities of DAWs, as well as for the processing and control of received/transmitted data from/to connected devices;
- an embedded feedback system to display information received from connected devices, for instance by means of visual, auditory, or haptic stimuli.

Smart instruments can be based on conventional acoustic instruments or be totally electronic. However, their features make them different from current augmented instruments or system for interactive performance: to the authors' best knowledge, none of the systems mentioned in Section 2 encompasses all the features listed above in a unique, playable, intelligent musical instrument.

One of the core characteristics that differentiate Smart Instruments from other interactive art systems is that they allow one to explore expressive and networked possibilities which would normally require the involvement of a multitude of equipment pieces: these include a musical instrument, a soundcard, a mixing interface, a computer, microphones, loudspeakers, a DAW and controller interfaces for it, as well as a networking system for local and remote communication. All these components are embedded in the instrument itself. In particular, Smart Instruments technology excludes the involvement of an external computation unit. Throughout many different forms of augmented, interactive or multimodal performance, the constant presence has been indeed a computation unit placed externally to the instrument (e.g., a laptop), which acted as the central hub for all the data to be processed. For example, it is a common setup to have two performers on stage, one playing an amplified acoustic instrument and the other having some sort of sensor apparatus which modulates the sound of that instrument. In this scenario, both performers have their sound go to a central computer which does the processing and then sends the results out of the house PA system. With the introduction of Smart Instruments all such computations and sound delivery are performed on the instrument itself.

In particular, whereas augmented instruments are mostly bespoke standalone systems, Smart Instruments are capable of directly exchanging musically relevant information between each other, not just in one direction and in a passive manner. In addition, Smart Instruments are capable of communicating with a diverse network of external devices connected to them. While augmented instruments capable of delivering multimodal information to external equipment exist (e.g., the augmented violin capable of generating real-time visuals related to bow movements described in [12]), Smart Instruments allow one additionally to receive, process, and display information to the player.

To achieve such a peer-to-peer communication, which is bidirectional and wireless, Smart Instruments can leverage both standard wireless networks technologies (e.g., Bluetooth, Wi-Fi, 4G) and ad-hoc ones (especially those allowing for a ultra-low latency transmission, which is a fundamental requisite for real-time applications in the musical domain).

3.2 Applications

The embedded intelligence of Smart Instruments allows for the delivery of musically relevant information to one or more Smart Instruments such as the notes played, the sensors values and their mappings to some sound effects parameters, or the generated sound. This information, for instance can be delivered in form of MIDI messages, Open Sound Control (OSC) messages, or audio signals. It can then be used to generate sounds that are reproduced directly on the receiving instrument thanks to its embedded DAW and sound delivery system and/or can be displayed by the instrument thanks to the embedded feedback system.

A variety of devices can be connected wirelessly to Smart Instruments, such as wearable technology (e.g., smart bracelets), smart phones, virtual reality headsets, or stage equipment such as lighting systems or smoke machines. Smart Instruments not only can deliver multimodal information to such devices in order to control their behaviour, but can also receive, process and display information coming from them.

The features of bidirectional, low-latency, and wireless

communication capabilities offered by Smart Instruments, as well as their embedded system for display feedback and sound delivery, enable an ecosystem of interoperable devices connecting performers as well as performers and audiences. This can take place not only in co-located, but also in remote settings. Such an ecosystem will make possible performer-performer and audience-performer interactions not offered by current augmented instruments. It can be exploited, for instance, for collaborative networked music creation, which has the potential to lead to novel forms of performance. Figure 1 illustrates an example of the data flow enabling such interactions and their human/machine agents.

A first example of the possible use cases implementing such interactions is represented by a novel form of jamming between players of such instruments: multiple players can wirelessly stream between each other and in real-time, audio content or musical messages (e.g., MIDI data) that are then reproduced by the sound delivery system of one or more receiving instruments. This is accomplished while the instruments themselves are being played by their performers. Moreover, each performer can control the mixing of the received audio streams. A second example consists of an enhanced creative content creation and delivery: performers interacting with the sensors embedded on their Smart Instruments not only can modulate the instruments sound production, but also deliver additional multimedia content to audience members in possession of smart devices. Such smart devices can produce multisensory feedback involving, for instance, visual, textual or tactile stimuli. A third example, consists of exploiting the feedback from the audience in a concert settings: information about body movements of each person in the audience are tracked by means of smart wearable devices, forwarded to a Smart Instrument, and used by its performer to modulate various aspects of the performance (e.g., the instrument timbre). Finally, a fourth example concerns remote rehearsals (at relatively close distances): the Smart Instruments of two or more performers can stream and receive in real-time the sounds generated by each of the performers, and the received audio stream is then reproduced and mixed directly by the instrument.

4. THE SENSUS SMART GUITAR

To date, a unique exemplar of musical instrument that encompasses all features of Smart Instruments exists: the Sensus Smart Guitar developed by the company MIND Music Labs [5] (see Figure 2). Such an instrument is based on a conventional acoustic guitar that is augmented with WSNs technologies. It is the result of the tight and interdisciplinary collaboration of instrument makers, software engineers, hardware engineers, sound designers, interaction designers, and IoT experts.

Sensus is built according to the crafting techniques of the most renowned of all the school of instrument making, that of the Stradivaris tradition [6] . All involved materials (wood, varnishes, etc.) are of high quality. Various parts of the instrument are made with a specific wood, carefully selected and naturally well seasoned. In particular, those woods include the special red spruce wood found in the Paneveggio Forest (in the Italian Dolomite mountains). This wood is characterized by a certain elasticity and particular honeycomb structure that allow the efficient transmission of sound waves and amplify sound.

In addition to regular knobs, switches and buttons, Sensus involves several sensors embedded in various parts of the instrument and ergonomically placed in order to not disrupt the natural interaction of the performer with the guitar. Specifically, these sensors include an inertial measurement units, five pressure sensors, two ribbon sensors, and an infrared proximity sensors. They allow for the tracking of a variety of gestures of the guitar player, including fingers pressure and position in various instrument areas (e.g., the neck), the distance of the hand from a specific part of the instrument located on the soundboard, and the position of the instrument (e.g., resulting from tilting up-down or front-back) and its linear acceleration along the three axes. As all augmented instruments, the tracked gestures are used to extend the expressive possibilities of the conventional acoustic guitar. In more detail, such gestures modulate the instrument sound and produce additional sounds thanks to a DAW running on a computation unit.

Such a computation unit is part of an embedded system, which is also responsible for the analog-to-digital conversion of sensors data and for the wireless connectivity. This system includes a multichannel soundcard and is powered by a battery that is also embedded in the instrument. The DAW employs a set of plugins for the processing of the guitar sound with a variety of effects, as well as for the generation of synthesized sounds by means of synthesizers and virtual instruments. It includes a loop station, and recording and playing features. It can be controlled via both MIDI and OSC messages.

Sensus can be connected to a regular PA system via standard jack cable and wirelessly. However, one of its main and peculiar features is that sounds, digitally processed or generated, can also be delivered by the instrument itself without the use of any external loudspeaker. This is achieved by means of a system of multiple actuators that transforms the instrument resonating wooden body into a 360° hi-fi loudspeaker. Such a system coupled with digital signal processing techniques, allows one to alter the timber of the instrument in manifold ways.

Furthermore, Sensus is equipped with bidirectional wireless connectivity leveraging both local networks and the Internet. This makes it possible the delivery and reception of different types of data from the instrument to a variety of smart devices (even including one or more Smart Guitars) and vice versa. Specifically, the connectivity technology includes Bluetooth Low Energy, standard Wi-Fi, and 4G. The data stream includes MIDI messages, OSC messages, and audio signals. The player's gestures tracked by the

[5] http://www.mindmusiclabs.com

[6] The traditional Stradivari's musical instruments craftsmanship in Cremona is inscribed on the Representative List of the Intangible Cultural Heritage of Humanity:
http://www.unesco.org/culture/ich/en/lists?RL=00719

Figure 1. A schematic representation of the bidirectional wireless connectivity between Smart Instruments and smart devices enabling new forms of interaction between performers and audience.

embedded sensors are also used to deliver and control such data stream.

The new forms of interaction between performers as well as between audience and performers enabled by this novel technology have started to be explored: MIND Music Labs has developed various applications running on smart devices that implement some of those interactions. One of these allows for novel forms of co-located jamming (i.e., collaborative and spontaneous music making). It runs on both Android- and iOS-based smartphones and tablets that wirelessly stream in real-time to Sensus audio content and/or musical messages (e.g., via MIDI or OSC). Such data are fed into the instrument DAW and then reproduced by its sound delivery system, while the instrument itself is being played by its performer. In turn, the performer acting on the instrument sensors can change the behaviour of the app running on one or more smart devices in possession of as many users (for instance, changing presets or the interface layout).

Thanks to its Internet connectivity feature, Sensus can easily share on various social networks audio content generated by playing on it and recorded in Hi-Fi quality. In addition, it can receive and reproduce audio signals streamed from remote repositories (e.g., songs streamed from Spotify), allowing a smart guitar player to play over them (e.g., for improvisation or rehearsing purposes).

Although Sensus is not yet on the market, to date various guitar players have had the chance to try it, both in public demos and during user experience experiments whose results have informed its ongoing development. In general, guitar players had positive feedback about their interaction with Sensus as far the sound production and modulation is concerned. The novel gesture-to-sound possibilities offered by the embedded sensors have been welcomed although, as one would expect for any augmented instrument, a general comment was that it takes time to learn, master, and incorporate them into the usual playing technique. One of the most appreciated features is the embedded actuation system, which allows one to play a guitar that vibrates like an acoustic instrument while incorporating effects like an electric one. Another feature which users appreciated is the presence of the embedded DAW and its continuous/discrete controls which are embedded in the instrument, since they eliminate the need for external equipment (e.g., footpedals) that a guitar player would usually have to carry, and that force him/her to a specific position in the stage. A more detailed description of the results of studies about the user experience during the interaction with Sensus is planned in another publication.

Videos of Sensus are available on the MIND Music Labs website[5] .

Figure 2. The Sensus Smart Guitar developed by MIND Music Labs.

5. DISCUSSION

Though augmented instrument research is well established, the potential designs and applications of Smart Instruments have only begun to be explored. Just as the transformative developments in mainstream computing have gradually shifted from individual devices to networked services, the most novel frontier to be explored in Smart Instruments is their capability to directly communicate with one another wirelessly and without a laptop as a mediator.

An open question is to what extent the presence of a laptop on stage affects the content of a performance. Many digital musical instruments use the laptop as a convenient source of computation without interacting with it directly in performance. Though this may seem to be aesthetically neutral, we suggest that there may be influences both overt and subtle from its presence. These include the tethering effect of cables, communication or audio processing latency, and questions of performer trust in a complex com-

puter system. Commonly used audio processing languages may also introduce subtle aesthetic biases in that certain musical outcomes are easier to achieve than others.

Therefore, the musical content and forms of interaction supported by direct instrument-to-instrument or instrument-to-audience communication may differ in significant and as-yet unforeseeable ways. To fully achieve this outcome, however, further development is needed on network technologies and protocols.

Current technological barriers that need to be addressed include network latency, which must meet strict requirements for musical performance. Wireless communication of electronic musical instruments has just recently seen an increase in popularity thanks to the use of dedicated communication protocols such as OSC. Nevertheless, such protocols are limited to the simple delivery of musical messages that are used to control the sound production of the receiving instrument. No IoT systems are currently available for the simultaneous, ultra-low latency, and bi-directional delivery of those control messages and of audio streams, which is an essential feature to enable novel forms of collaborative music creation. Existing systems are typically wired, which is in contrast with the seamless and ubiquitous connection needs of modern Internet wireless technologies. The main challenge in exchanging audio signals resides in achieving a low-latency bidirectional communication over wireless networks. This implies the creation of a technological infrastructure that is capable to transmit audio content from one musician to another not only in hi-fi quality, but also with a negligible amount of delay, e.g., in order to allow performers to play in synchronous ways. Current technologies do not satisfy these tight constraints needed for the real-time transmission of audio content both at short and at large distances. Indeed, while the most cutting-edge current networks can deliver very high data rates, they are also restricted by communication delays of the order of 25ms [40]. This delay is unacceptable for real-time collaborative music.

The introduction of the proposed novel class of musical instruments opens questions regarding their standardization. The authors consider the features listed in section 3.1 as the minimum required for a musical instrument to be considered a Smart Instrument. However, as any process related to the development of a technical standard, the standardization of Smart Instruments will ultimately be the result of a community effort. It is the authors' hope that the present work could also serve to foster a discussion towards such a topic.

Finally, it is worth considering the aesthetic opportunities and pitfalls that a new generation of Smart Instruments will produce. Smart Instruments have the potential to add many new dimensions of control and communication onto existing performances. On the other hand, live multimodal and augmented performances have a multi-decade history, and as Tanaka observed in 2000 [41], more is not always better: "Discussions of computer based instruments often tend to focus on the power or capability of the instrument. With sensor instruments the question typically raised is how many synthesis parameters it allows the musician to control.... While this may show off the power of computers to simultaneously control multiple parameters across multiple media, it does little to convey the expression of the performer. Instead, viewing the situations from the standpoint of creative applications of limitations may yield more musical results." Whatever the capabilities of future Smart Instruments, thoughtful applications of them should retain the focus on the expression of the human performer, with an understanding that one or two well-chosen control dimensions may be worth more than the most extensive possible performance environment.

Acknowledgments

The authors wish to thank all MIND Music Labs team members. The work of Carlo Fischione is sponsored by the TNG SRA ICT project "TouCHES".

6. REFERENCES

[1] T. Machover and J. Chung, "Hyperinstruments: Musically intelligent and interactive performance and creativity systems," in *Proceedings of the International Computer Music Conference*, 1989.

[2] E. R. Miranda and M. M. Wanderley, *New digital musical instruments: control and interaction beyond the keyboard*. AR Editions, Inc., 2006, vol. 21.

[3] G. Weinberg, "Interconnected musical networks: Toward a theoretical framework," *Computer Music Journal*, vol. 29, no. 2, pp. 23–39, 2005.

[4] L. Atzori, A. Iera, and G. Morabito, "The internet of things: A survey," *Computer networks*, vol. 54, no. 15, pp. 2787–2805, 2010.

[5] T. Meurisse, A. Mamou-Mani, R. Causse, B. Chomette, and D. B. Sharp, "Simulations of modal active control applied to the self-sustained oscillations of the clarinet," *Acta Acustica united with Acustica*, vol. 100, no. 6, p. 1149–1161, 2014.

[6] T. Meurisse, A. Mamou-Mani, R. Caussé, B. Sluchin, and D. B. Sharp, "An active mute for the trombone," *The Journal of the Acoustical Society of America*, vol. 138, no. 6, pp. 3539–3548, 2015.

[7] J. A. Paradiso, "Electronic music: new ways to play," *IEEE Spectrum*, vol. 34, no. 12, 1997.

[8] D. Overholt, E. Berdahl, and R. Hamilton, "Advancements in actuated musical instruments," *Organised Sound*, vol. 16, no. 02, pp. 154–165, 2011.

[9] S. Jordà, "Instruments and players: Some thoughts on digital lutherie," *Journal of New Music Research*, vol. 33, no. 3, pp. 321–341, 2004.

[10] F. Bevilacqua, N. Rasamimanana, E. Fléty, S. Lemouton, and F. Baschet, "The augmented violin project: research, composition and performance report," in *Proceedings of the International Conference on New Interfaces for Musical Expression*, 2006, pp. 402–406.

[11] D. Overholt, "Violin-related HCI: A taxonomy elicited by the musical interface technology design space," in *Arts and Technology*. Springer, 2012, pp. 80–89.

[12] L. S. Pardue, C. Harte, and A. P. McPherson, "A low-cost real-time tracking system for violin," *Journal of New Music Research*, vol. 44, no. 4, 2015.

[13] J. Thibodeau and M. M. Wanderley, "Trumpet augmentation and technological symbiosis," *Computer Music Journal*, vol. 37, no. 3, 2013.

[14] N. Bouillot, M. Wozniewski, Z. Settel, and J. R. Cooperstock, "A mobile wireless augmented guitar." in *Proceedings of the International Conference on New Interfaces for Musical Expression*, 2008, pp. 189–192.

[15] O. Lähdeoja, M. M. Wanderley, and J. Malloch, "Instrument augmentation using ancillary gestures for subtle sonic effects," *Proceedings of the Sound and Music Computing Conference*, pp. 327–330, 2009.

[16] O. Lahdeoja, "An augmented guitar with active acoustics," in *Proceedings of the Sound and Music Computing Conference*, 2015.

[17] A. McPherson, "Buttons, handles, and keys: Advances in continuous-control keyboard instruments," *Computer Music Journal*, vol. 39, no. 2, pp. 28–46, 2015.

[18] L. Turchet, "The Hyper-Hurdy-Gurdy," in *Proceedings of the Sound and Music Computing Conference*, 2016.

[19] N.-W. Gong, N. Zhao, and J. A. Paradiso, "A customizable sensate surface for music control." in *Proceedings of the International Conference on New Interfaces for Musical Expression*, vol. 12, 2012, pp. 417–420.

[20] D. Newton and M. T. Marshall, "Examining how musicians create augmented musical instruments," in *Proceedings of the International Conference on New Interfaces for Musical Expression*, 2011.

[21] E. Berdahl and W. Ju, "Satellite CCRMA: A musical interaction and sound synthesis platform," in *Proceedings of the International Conference on New Interfaces for Musical Expression*, 2011.

[22] A. Refsum Jensenius and V. Johnson, "Performing the electric violin in a sonic space," *Computer Music Journal*, vol. 36, no. 4, pp. 28–39, 2012.

[23] E. J. Berdahl, "Applications of feedback control to musical instrument design," Ph.D. dissertation, Stanford University, 2010.

[24] E. Berdahl, S. Backer, and J. Smith, "If I had a hammer: Design and theory of an electromagnetically prepared piano," in *Proceedings of the International Computer Music Conference*, 2005, pp. 81–84.

[25] A. McPherson and Y. Kim, "Augmenting the acoustic piano with electromagnetic string actuation and continuous key position sensing," in *Proceedings of the International Conference on New Interfaces for Musical Expression*, 2010.

[26] D. Rector and S. Topel, "EMdrum: an electromagnetically actuated drum," in *Proceedings of the International Conference on New Interfaces for Musical Expression*, 2014.

[27] N. C. Britt, J. Snyder, and A. McPherson, "The EMvibe: an electromagnetically actuated vibraphone," in *Proceedings of the International Conference on New Interfaces for Musical Expression*, 2012.

[28] G. Shear and M. Wright, "The electromagnetically sustained Rhodes piano," in *Proceedings of the International Conference on New Interfaces for Musical Expression*, 2011.

[29] H. Boutin and C. Besnainou, "Physical parameters of the violin bridge changed by active control," *Journal of the Acoustical Society of America*, vol. 123, no. 5, p. 3656, 2008.

[30] A. Zoran and J. A. Paradiso, "The chameleon guitar-guitar with a replaceable resonator," *Journal of New Music Research*, vol. 40, no. 1, pp. 59–74, 2011.

[31] K. Buys, D. Sharp, and R. Laney, "Developing a hybrid wind instrument: using a loudspeaker to couple a theoretical exciter to a real resonator," in *Proceedings of the International Symposium on Musical Acoustics*, 2014, pp. 331–336.

[32] E. Berdahl, "How to make embedded acoustic instruments." in *Proceedings of the International Conference on New Interfaces for Musical Expression*, 2014, pp. 140–143.

[33] A. McPherson and V. Zappi, "An environment for Submillisecond-Latency audio and sensor processing on BeagleBone black," in *Audio Engineering Society Convention 138*. Audio Engineering Society, 2015.

[34] A. Zoran, "The 3d printed flute: digital fabrication and design of musical instruments," *Journal of New Music Research*, vol. 40, no. 4, pp. 379–387, 2011.

[35] D. Overholt, "The Overtone Fiddle: an actuated acoustic instrument," in *Proceedings of the International Conference on New Interfaces for Musical Expression*, 2011.

[36] W. W. Dargie and C. Poellabauer, *Fundamentals of wireless sensor networks: theory and practice*. John Wiley & Sons, 2010.

[37] A. Willig, "Recent and emerging topics in wireless industrial communication," *IEEE Transactions on Industrial Informatics*, vol. 4, no. 2, pp. 102–124, 2008.

[38] *IEEE Std 802.15.4-2996, September, Part 15.4: Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Low-Rate Wireless Personal Area Networks (WPANs)*, IEEE, 2006.

[39] A. J. G. X. Liu, "Cross-layer Design of Distributed Control over Wireless Network," *Systems and Control: Foundations and Applications, (Ed. T. Basar), Birkhauser*, 2005.

[40] G. P. Fettweis, "The tactile internet: applications and challenges," *IEEE Vehicular Technology Magazine*, vol. 9, no. 1, pp. 64–70, 2014.

[41] A. Tanaka, "Musical performance practice on sensor-based instruments," *Trends in Gestural Control of Music*, vol. 13, pp. 389–405, 2000.

EXPRESSIVE HUMANOID ROBOT FOR AUTOMATIC ACCOMPANIMENT

Guangyu Xia[1], Mao Kawai[2], Kei Matsuki[2], Mutian Fu[1], Sarah Cosentino[2], Gabriele Trovato[2], Roger Dannenberg[1], Salvatore Sessa[2], Atsuo Takanishi[2]

[1] Carnegie Mellon University, [2]Waseda University
{gxia, mutianf, rbd}@andrew.cmu.edu[1]
contact@takanishi.mech.waseda.ac.jp[2]

ABSTRACT

We present a music-robotic system capable of performing an accompaniment for a musician and reacting to human performance with gestural and facial expression in real time. This work can be seen as a marriage between social robotics and computer accompaniment systems in order to create more musical, interactive, and engaging performances between humans and machines. We also conduct subjective evaluations on audiences to validate the joint effects of robot expression and automatic accompaniment. Our results show that robot embodiment and expression improve the subjective ratings on automatic accompaniment significantly. Counterintuitively, such improvement does not exist when the machine is performing a fixed sequence and the human musician simply follows the machine. As far as we know, this is the first interactive music performance between a human musician and a humanoid music robot with systematic subjective evaluation.

1. INTRODUCTION

In order to create more musical, interactive, and engaging performances between humans and machines, we contribute the first automatic accompaniment system that reacts to human performance with humanoid robot expression (as shown in Figure 1). This study bridges two existing fields: *social robotics* and *automatic accompaniment*.

Figure 1. The robotic automatic accompaniment system.

On one hand, score following and automatic accompaniment systems (often briefly named automatic accompaniment) have been developed over the past 30 years to serve as virtual musicians capable of performing music with humans. Given a performance reference (usually a score representation), these systems take human performance as an input, match the input to the reference, and output the accompaniment by adjusting its tempo in real time. The first systems invented in 1984 [1][2] used simple models to anticipate the tempo of a monophonic input. Ever since then, many studies extended the model to achieve more expressive music interactions. These extensions include polyphonic [3] and embellished [4] input recognition, smooth tempo adjustment [5][6], and even expressive reaction with music nuance [7]. While most efforts focused on the system's auditory aspects, two major issues of automatic accompaniment remain unexplored. First, no model has considered the virtual musician's gestural and facial expressions, despite the fact that visual cues also serve as an important part of music interaction [8][9]. Second, no subjective evaluation has been conducted to validate that automatic accompaniment is a better solution than fixed media for human-computer music performance.

On the other hand, social robots have been developed to interact with humans or other agents following certain rules of social behaviors. Many studies have shown that robot expression, especially humanoid expression, significantly increases the engagement and interaction between humans and computer programs in many forms, such as telecommunication [10] and dialog systems [11]. However, music interaction, as high-level social communication, has not been paid much attention in this context. Though we have seen the development of several music robots, none are able to react to other musicians with human-like expression yet.

It is clear to see that automatic accompaniment and social robotics can complement each other. Therefore, we integrated the saxophonist robot developed at Waseda University into an existing framework of automatic accompaniment. To be specific, the system currently takes a human musician's MIDI flute performance as input and outputs acoustic accompaniment with gestural and facial expression. The (larger scale) gestural expression reacts to music phrases while the (smaller scale) facial expression reacts to local tempo changes. Of course, our first integration does not consider *all* aspects of gestural and facial expression. The current solution considers body and eyebrow movements, and we believe that other aspects of expression can be processed in a similar way.

In addition, we conducted subjective evaluations of this integration on audiences to validate the joint effects of robot expression and automatic accompaniment. Our hypothesis is that with humanoid robot expression, an automatic accompaniment system provides more musical, interactive, and engaging performance between humans and machines. We showed video clips in different conditions (with/without expression, with/without accompaniment) to audiences and used repeated-measure ANOVA to measure the difference between different conditions. Our results show that robot embodiment, especially facial expression, improves the subjective ratings on automatic accompaniment significantly. Counterintuitively, such improvement does not exist when the machine is performing a fixed media and the human musician simply follows the machine.

The next section presents related work. Section 3 presents the design of our saxophone robot with a focus on its control of body and eyebrow movements. Section 4 shows the automatic accompaniment framework with a focus on the mapping from MIDI performance to robot motions. In Section 5, we present the subjective evaluations and the experimental results.

2. RELATED WORK

Related work in music robotics can be categorized according to two perspectives: non-humanoid vs. humanoid, and pre-programmed vs. and interactive. Our study considers interactive humanoid robot.

2.1 Non-humanoid vs. Humanoid Music Robots

Musical player robots play an important role in the study of musical interaction. Non-humanoid music player robots with advanced interaction capabilities, such as Shimon [12] and Haile [13], have been used extensively to test pure musical interaction models. On the other hand, humanoid robots can be used as a tool for the validation of *embodied* interaction models. We believe that non-verbal gestures can be mimicked exquisitely and replicated by robots to study the influence of embodiment in musical interaction.

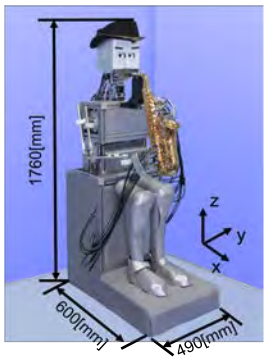
2.2 Pre-programmed vs. Interactive Music Robots

While most music robots have the potential to adapt their performance to others, most of their performances are still pre-programmed. However, we started to see more interactive music robots developed in the past decade. Generally, these robots detect beats from music and adjust their behaviors to stay synchronized with the music. These systems include interactive dancers [14], a Theremin player [15], singers [16], drummers [17], marimba players [12], and other percussion players [18]. However, very few of them react to music with gestural expression or have been evaluated experimentally by human subjects. As far as we know, the only subjective evaluation for interactive music robots was done in the work on Shimon [12]. This work showed that the visual contact with the marimba robot improves audiences' subjective ratings. On top of this, our study incorporates humanoid gestural and facial expression and conducts the first evaluation to inspect the joint effect of robot expression and music interaction.

3. HUMANOID SAXOPHONIST ROBOT

3.1 WAseda Saxophonist Robot (WAS)

The development of WAseda Saxophonist Robot (WAS) [19][20] was started in 2008. The robot was designed with a critical focus on the physiology and anatomy of the human organs involved during saxophone playing. The fourth version of the robot (WAS-4) was completed in 2015. Face and trunk mobility has been increased, to add basic interaction abilities during artistic joint performances with human partners. During joint musical performances, in fact, musicians cannot use vocal signs and must rely on non-verbal body communication for synchronization. The robot is now able to perform human-like non-verbal signaling, giving partner human players real-time cues on its interpretation, allowing for a better control over synchronization and improving the interaction experience as well as the overall joint musical performance. Figure 2 shows the general design of the robot used in this study.



Function	Body Parts	DoFs
Sound production	Lips	2
	Oral cavity	1
	Tongue	1
	Lung	Pump 1 Valve 1
Key stroke	Fingers	Left 8 Right 11
Body movement	Hip	1
Facial expression	Eyebrow	1
Total		29

Figure 2. The design of Waseda saxophonist robot (WAS).

3.2 Body and Eyebrow Movements

The two interactive movements used in this study are: swinging the upper body and raising/frowning eyebrows. The body positions during a swing movement are shown in Figure 3, where the robot starts from a neutral position (left), swings forward and backward (middle two snapshots), and finally comes back to the neutral position again (right). Eyebrow movements are illustrated in Figure 4, where the left one is neutral, the middle one is raised, and the right one is frowning.

Figure 3. An illustration of the four positions in a body movement.

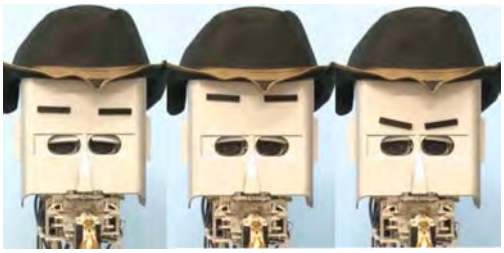


Figure 4. An illustration of the three eyebrow positions.

4. AUTOMATIC ACCOMPANIMENT WITH ROBOT EXPRESSION

This section describes how the robot reacts to human performance. There are three main steps: score following, tempo estimation, and the mapping from tempo to robot expression. The logic flow is shown in Figure 5. Again, the current system takes a human's monophonic MIDI flute performance as input and outputs acoustic accompaniment with eyebrow and body swing movements.

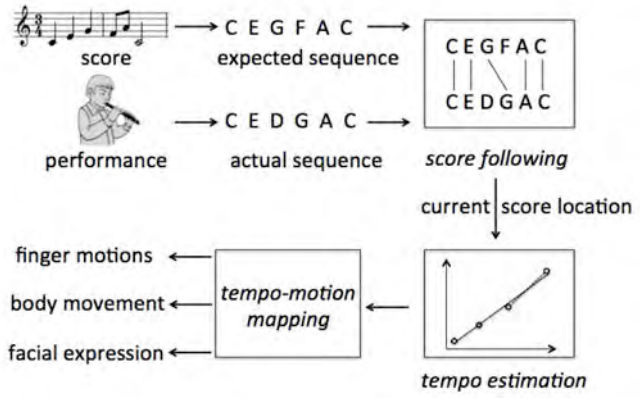


Figure 5. A system diagram of automatic accompaniment system with robot expression.

4.1 Score Following

Given the human performance, the first step of the process is score following, which keeps track of the current score location by finding the best match between score and performance. In our case, both score and performance are represented by a sequence of pitch symbols, with performance being the actual sequence and the score being the expected sequence. If the performance exactly follows the order of the score, we would simply update score location stepwise. However, since human performance will add and skip notes, we need an online matching algorithm. The current system adopts the solution introduced in [1], which first computes the "matched length" associated with each performance note and then updates the score location *only* when this length exceeds previous reported ones. Formally, the matched length is computed by:

$$MatchedLength = \#\text{ matched note} - \#\text{ skipped note} \quad (1)$$

Here, # represents the number of elements. Figure 6 shows an example, where the first line is the score, second line is the performance, and the third line is the matched length associated with each performed note. In this example, the algorithm reports a match and updates the score location after C, E, G, A, C are performed, since their matched lengths exceed previous ones.

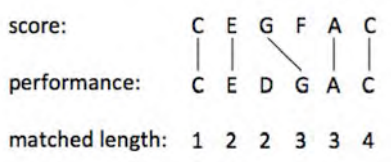


Figure 6. An illustration of the score following algorithm with one added and one skipped notes.

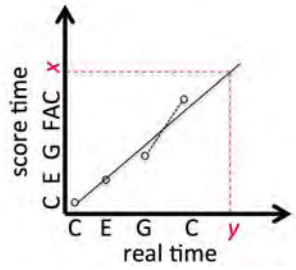


Figure 7. An example of the tempo estimation algorithm.

4.2 Temp Estimation

Given the matching results of score following, tempo estimation quantifies how fast/slow the human performance is against the timings specified in the score. This result will be later used to control the robot reactions. We adopt a "performance-score timing" 2-D representation (as shown in Figure 7) and represent tempi as slopes on this 2-D plane. The unit of score time is *beat*, the unit of performance time is *second,* and hence the unit of tempo is *beats per second*. We estimate tempo in two scales: micro and macro. The former is based on two adjacent notes, while the latter is based on the notes within a 4-beat interval.

Formally, let the matched notes reported by score following be $m = [m_1, m_2, \ldots, m_i, \ldots]$. Also, let the corresponding performance time and score time be $p = [p_1, p_2, \ldots, p_i, \ldots]$ and $s = [s_1, s_2, \ldots, s_i, \ldots]$, respectively. Then, the micro-scale tempo is defined as $v = [v_1, v_2, \ldots, v_i, \ldots]$, where

$$v_i = \begin{cases} (s_i - s_{i-1})/(p_i - p_{i-1}), & i > 1 \\ 1, & i = 1 \end{cases} \quad (2)$$

The macro-scale tempo is defined as $V = [V_1, V_2, \ldots, V_i, \ldots]$. If there are n matched notes within the score time interval of $[s_i - 4, s_i]$, then V_i is computed via the method of least squares:

$$V_i = \begin{cases} \frac{\sum_{j=i-n+1}^{i}(p_j - \bar{p})(s_j - \bar{s})}{\sum_{j=i-n+1}^{i}(p_j - \bar{p})}, & n > 1 \\ 1, & n = 1 \end{cases} \quad (3)$$

Here,

$$\bar{p} = \frac{1}{n}\sum_{j=i-n+1}^{i} p_j \text{ and } \bar{s} = \frac{1}{n}\sum_{j=i-n+1}^{i} s_j \quad (4)$$

Figure 7 shows an example of tempo estimation corresponding to the score following example in Figure 6, where the solid line represent the macro-scale tempo of the last matched note C, and the dotted line represents its micro-scale tempo. (Note that we do not estimate the tempi of unmatched notes.)

4.3 Mapping from Tempo to Robot Motions

We designed rule-based methods to control the robot motion by the estimated tempo. The current system separates robot motions into three groups: finger motions which are controlled by macro-scale tempo, body movements which are controlled by the deviation of macro-scale tempo, and eyebrow movements which are controlled by the deviation of micro-scale tempo. These rules are designed according to domain knowledge of music performance.

4.3.1 Finger Motions

Finger motions control the accompaniment, whose timings are specified in another pre-defined score that is synchronized with the score for human performance. The robot uses the latest macro-scale tempo estimation and extrapolates this tempo (slope) to estimate and schedule the next note. Figure 7 shows an example, where the nearest accompaniment note after the last human performed note C is at beat x, and its actual performance time will be y. It is important to notice that finger motions requires high timing accuracy, but robot mechanics has unavoidable latency. To overcome the latency, we schedule the notes ahead of their estimated onset times. Formally, if the latency for the MIDI flute played by the human performer is l_1 and the latency for the robot fingers is l_2, notes whose estimated time is t will be scheduled to execute at $t' = t - (l_2 - l_1)$. In practice, t' is around 40 milliseconds. (We point readers to [6] for more details on adjusting latency in real time performance.)

4.3.2 Body Movements

Body movements are controlled by the deviation of the macro-scale tempo. If the two latest estimated macro-scale tempi both speed up/slow down beyond a certain threshold, a body movement is triggered. By referring to the notations in the last section, for $i > 1$, a body movement is triggered if:

$$\left|\frac{v_{i-j}-v_{i-j-1}}{v_{i-j}}\right| > p \text{ for } j = 0 \text{ and } 1 \quad (5)$$

The rationale of this rule is that performers often use body movements to indicate smooth tempo changes. The current system sets p = 5%. Besides this rule, we also insert a body movement at the beginning and the ending of the robot performance.

4.3.3 Eyebrow Motions

Eyebrow motions are controlled by the deviation of the micro-scale tempo. If the two latest estimated micro-scale tempi both slow up beyond a certain threshold, both eyebrows will raise. Similarly, if the tempi speed up beyond a certain threshold, a *frown* motion is triggered. If none of these two conditions are met, eyebrows stay at the neutral position. Formally, for $i > 1$, and $j = 0$ and 1,

$$\text{eyebrow motion} = \begin{cases} \text{raise}, & \text{if } \frac{v_{i-j}-v_{i-j-1}}{v_{i-j}} > q \\ \text{frown}, & \text{if } \frac{v_{i-j}-v_{i-j-1}}{v_{i-j}} < -q \\ \text{neutral}, & \text{otherwise} \end{cases} \quad (6)$$

The rationale of this rule is that eyebrow motions are often associated with sudden tempo changes. The current system sets q = 5%. Note that eyebrow motions will be more frequent than body movements under the same threshold because micro-scale tempi are more sensitive.

5. SUBJECTIVE EVALUATION

We conducted subjective evaluations on audiences to validate the effects of robot expression. We first inspected whether the robot helps with automatic accompaniment. Then, we inspected whether the robot helps with fixed media performance (in which the robot plays a fixed performance and the human performer has to adapt to the robot). Finally, we compared these two results to see the joint effect of robot expression and automatic accompaniment.

5.1 The Robot Effect on Automatic Accompaniment

Our hypothesis is that with humanoid robot expression, the automatic accompaniment system provides more musical, interactive, and engaging performance between humans and machines. To test this claim, we recorded videos of human-computer interactive performances (of the same piece of music) in 3 different conditions of robot embodiment and invited audiences to provide subjective ratings on these videos.

5.1.1 Video Recording Setup

The videos were recorded as shown in Figure 8, with the human performer and the robot standing opposite to each other. Figure 1 shows a corresponding screen shot.

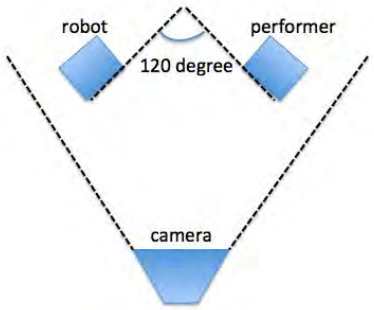


Figure 8. The layout of the video recording setup.

Index	Robot setting
A	Blocked robot
B	Static body
C	Full expression

Table 1. The three different conditions for robot setting.

5.1.2 Conditions of Robot Embodiment

The 3 performance conditions are listed in Table 1, where higher index corresponds to greater functionality of the robot. Note that in condition A (blocked robot), we put a cover in front of the robot so that neither the human performer nor the camera could see the robot. The purpose was to block the visual cues but retain the same sound source. In condition B (static body), the robot's body and eyebrows do not move; the only working parts are the mouth and fingers. In condition C (full expression), the robot movements include mouth, fingers, body, and eyebrows.

5.1.3 Survey

We showed the recorded performance videos in all 3 conditions in a random order to each audience subject without directly revealing the condition. Each video is about 80 seconds long. After each video, audiences were asked to rate the performance according to three criteria:

1. *Musicality:* how musical the performance was.
2. *Interactivity:* how close the interaction was between the human performer and the machine.
3. *Engagement:* how engaged the human performer was.

For all 3 criteria, we used a 5-point Likert scale from 1 (very low) to 5 (very high).

5.1.4 Hypothesis Test

The null hypothesis is that different conditions have no effect on automatic accompaniment and therefore the ratings under different conditions are the same. Formally:

$$H_0: \quad \mu_A = u_B = \mu_C \tag{7}$$

Similarly, the alternative hypothesis is that:

$$H_1: \quad \exists i,j \in \{A,B,C\}: \mu_i \neq \mu_j \tag{8}$$

Since all the subjects experienced all the conditions, we used within-subject ANOVA [21] (also known as repeated measurement study) to compute the mean standard error (MSE) and p-value. We used the Huynh-Feldt correction [21] when the sphericity of the data is not met.

5.1.5 Results

A total of n = 33 subjects (14 female and 19 male) have completed the survey. The aggregated result (as in Figure 9) shows that the robot effect improves the subjective ratings of automatic accompaniment.

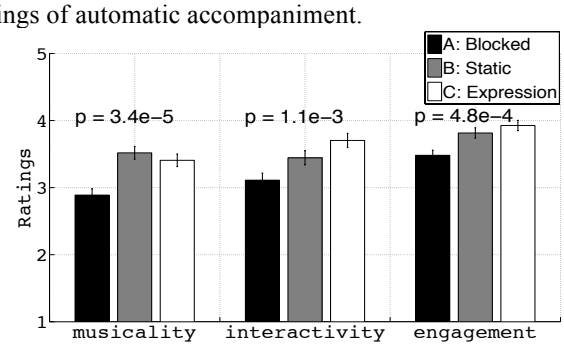


Figure 9. The subjective evaluation results of the robot effect on automatic accompaniment.

Here, different colors represent different conditions. The heights of the bars represent the means of the ratings and the error bars represent the MSEs. It is clear that the robot embodiment and expression improves the ratings and such improvements are monotonic (except for *musicality*) when the functionality of the robot increases. For all three criteria, the p-values are much smaller than 0.005 and hence the improvements are statistically significant.

5.2 The Robot Effect on Fixed Media Performance

In addition to the robot effect on automatic accompaniment, we also inspected whether the robot helps with fixed media performance. In this case, the robot played a pre-recorded performance and the human musician adapted to the robot. Similar to Section 5.1, the null hypothesis is that different conditions have no effect on the subjective ratings of fixed media performance. With exactly the same video recording setup, conditions of robot setting, and survey process, the result (as in Figure 10) shows that robot embodiment and expression do not help with fixed media performance.

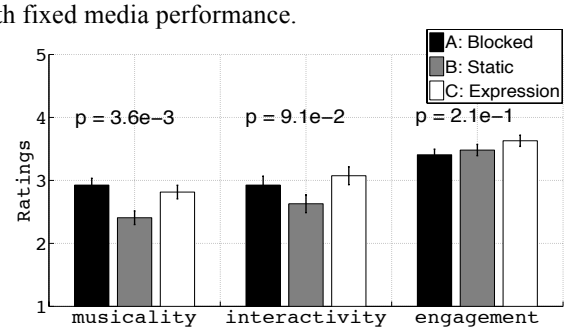


Figure 10. The subjective evaluation results of the robot effect on fixed media performance.

Counterintuitively, for musicality the robot decreases the ratings with the p-value smaller than 0.005. For interactivity and engagement, though we see evidence of improvement, the associated p-values are both larger than 0.05.

5.3 A Comparison between Automatic Accompaniment and Fixed Media Performance.

We finally inspect the joint effect of automatic accompaniment and the robot effect by putting the results of Figure 9 and Figure 10 together, as shown in Figure 11.

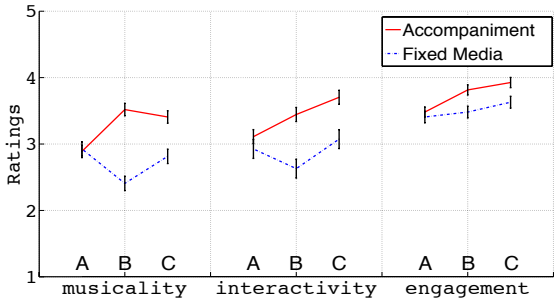


Figure 11. The joint effect of automatic accompaniment and robot expression.

For all 3 criteria, while the difference between automatic accompaniment and fixed media is not significant (p-value > 0.05) in condition A, the difference becomes much more significant (p-value < 0.005) in conditions B and C. This result suggests that neither automatic accompaniment nor robot expression alone is significantly better than fixed media performance. Only when we combine these two factors, does the music performance between the human and the machine become significantly more musical, interactive, and engaging.

6. CONCLUSIONS AND FUTURE WORK

In conclusion, we have combined the efforts of social robotics and automatic accompaniment to create the first automatic accompaniment system with humanoid robot expression. Our subjective evaluation shows that expressive humanoid robots lead to more musical, interactive,

and engaging automatic accompaniment. Counterintuitively, this effect does not exist for fixed media performance. This study contributes to the computer music community by providing the first subjective evaluation on automatic accompaniment and its joint effect with robot expression. It also contributes to the social robotics community by proving that the benefits of humanoid robots generalize to the interactive music performance scenario. The result shows the benefit of combining interactive computer music system with humanoid robot, which points to the integration of these two fields for future research.

In the future, we would like to continue this study in the following three directions:

Visual cues: The current robot is still blind. We are going to place cameras on the robot and use visual cues to guide the generation of robot expression.

Learning-based robot expression: So far, the robot expression is generated by a rule-based method. To make the method learning-based, we are going to use a motion capture system to collect rehearsal videos with facial and gestural expression.

Evaluation of the performance experience: So far, the subjective evaluation is conducted on audiences only. We are going to invite multiple performers as our subjects in the future.

Acknowledgments: We thank to the General Directorate for Cultural Promotion and Cooperation for its support to RoboCasa, and we also thank to SolidWorks Corp for their support to the research.

7. REFERENCES.

[1] R. Dannenberg, “An online algorithm for real-time accompaniment,” in Proceedings of the International Computer Music Conference, 1984, pp. 193-198.

[2] B. Vercoe, “The synthetic performer in the context of live performance,” in Proceedings of the International Computer Music Conference, 1984, pp. 199-200.

[3] J. Bloch and R. Dannenberg, “Real-time accompanyment of polyphonic keyboard performance,” in Proceedings of the International Computer Music Conference, 1985, pp. 279-290.

[4] R. Dannenberg, and H. Mukaino, “New techniques for enhanced quality of computer accompaniment,” in Proceedings of the International Computer Music Conference, 1988, pp. 243–249.

[5] A. Cont., “ANTESCOFO Anticipatory Synchronization and Control of Interactive Parameters,” In Computer Music Proceedings of International Computer Music Conference, 2008.

[6] D. Liang, G. Xia, and R. Dannenberg, “A framework for coordination and synchronization of media,” in Proceedings of the New Interfaces for Musical Expression, 2011.

[7] G. Xia and R. Dannenberg, “Duet Interaction: learning musicianship for automatic accompaniment,” in Proceedings of the International Conference on New Interfaces for Musical Expression, 2015.

[8] K. Katahira et al., “The role of body movement in co-performers’ temprral coordination,” in Proceedings of ICoMCS, 2007, pp. 72.

[9] S. Kawase, “Importance of communication cues in music performance according to performers and audience,” International Journal of Psychological Studies, 2014, pp. 49-64.

[10] S. Adalgeirsson, “Mebot, a robotic platform for socially embodied telepresence,” in Proceedings of the fifth ACM/IEEE International Conference on Human-Robot Interaction (HRI-10), ACM/IEEE International, 2010, pp. 15-22.

[11] J.K. Lee and C. Breazeal, “Human social response toward humanoid robot’s head and facial features,” CHI Extended Abstracts, 2010, pp. 4237-4242

[12] G. Hoffman, and G. Weinberg, "Interactive improvisation with a robotic marimba player," in Autonomous Robots 31, no. 2-3, 2011, pp. 133-153.

[13] S. Sun, M. Trishul, and G. Weinberg. "Effect of Visual Cues in Synchronization of Rhythmic Patterns." In Proceedings of International Conference of Music Perception and Cognition, 2012.

[14] G. Xia, et al., "Autonomous robot dancing driven by beats and emotions of music," In Proceedings of the 11th International Conference on Autonomous Agents and Multiagent Systems. 2012, pp. 205-212.

[15] A. Lim, et al., "Robot musical accompaniment: integrating audio and visual cues for real-time synchronization with a human flutist," IEEE International Conference on Intelligent Robots and Systems, 2010, pp. 1964-1969.

[16] T. Otsuka, et al., "Incremental polyphonic audio to score alignment using beat tracking for singer robots," IEEE/RSJ International Conference on. IEEE, 2009, pp. 2289-2296.

[17] G. Weinberg, S. Driscoll, and M. Parry, “Musical interactions with a perceptual robotic percussionist,” in IEEE International Workshop On Robot and Human Interactive Communication, 2005, pp. 456-461.

[18] A. Kapur, “Multimodal techniques for human/robot Interaction.” In Musical Robots and Interactive Multimodal Systems, Springer Berlin Heidelberg, 2011, pp. 215-232.

[19] J. Solis, T. Ninomiya, K. Petersen, M. Takeuchi, and A. Takanishi, “Development of the anthropomorphic saxophonist robot WAS-1: Mechanical design of the simulated organs and implementation of air pressure feedback control”. In Advanced Robotics, 24(5-6), 2010.

[20] K. Matsuki, K. Yoshida, S. Sessa, S. Cosentino, K. S. Kamiyama, and A. Takanishi “Facial Expression Design for the Saxophone Player Robot WAS-4”, In proceedings of the 21st CISM IFToMM Symposium on Robot Design, Dynamics and Control, 2016.

[21] R. Ellen and E. Girden. “ANOVA: Repeated measures”. No. 84. Sage, 1992.

List of Authors