A novel audio-to-score alignment method using velocity-driven DTW

Oriol Romaní Picas

MASTER THESIS UPF / 2014
Master in Sound and Music Computing

Master thesis supervisor:
Dr. Julio Carabías / Dr. Jordi Janer
Department of Information and Communication Technologies
Universitat Pompeu Fabra, Barcelona

Abstract

The problem of automatic audio to score alignment is nowadays well understood, leading to high accuracies even for polyphonic music signals with several instruments playing at the same time. Traditionally, the evaluation metrics rely on the distance between the ground truth and the estimated note onsets, considering a fixed tolerance threshold (e.g. 200 ms). This criterion is suitable for many applications such as page turning or informed sound source separation. However, other applications as automatic musical accompaniment require more advance alignment, considering musical aspects such as tempo in order to control the time differences between onsets of consecutive notes, leading to aligned scores with more musical meaning.

The aim of this master thesis is to provide a solution to guide the alignment process from a more musical point of view in order to implement an automatic musical accompaniment system. To this end, the most reliable score follower algorithm of the MIREX competition is used as a base line and extended taking into account the musical restrictions. The proposed method is based on Dynamic Time Warping (DTW), where a tempo controller is enforced within the minimum cost path computation in a novel way.

Acknowledgements

I whish to thank my supervisors Jordi Janer and Julio Carabias for their guidance and support. I specially want to thank Julio for going beyond the duties of a supervisor and also for his extensive knowledge, patience and kindness. I would also like to thank Xavier Serra for giving me the opportunity of being part in the SMC Master that has been the great academic experience in my live. Also all the members of the MTG and specially the most attentive teachers to me; Emilia Gómez, Agustín Martorell, Waldo Nogueira, Sergi Jordà, Mohamed Sordo and Enric Giné. I thank my fellow students and specially the ones that have helped me more; Úrbez Capablo, Nadine Kroher, Jakue López, Aram Estiu, Roger Rios, Giacomo Herrero, Francisco Rodríguez, Dara Dabiri and Hector Parra. Also all the others for such a great time in Barcelona. Finally I would like to thank Caterina for her patience and understanding; this would have not been possible without you.

Contents

List of Figures

List of Tables

Chapter 1

INTRODUCTION

1.1 Motivation

Real music performances tend to be irregular in time due to the use of tempo as an expressivity facet. In the growing research field of Music Information Retrieval (MIR) most of the research problems need a previous alignment of the audio coming from the performance and some metadata like the score or the lyrics. There are a large amount of approaches to automatically align two time series but for some particular applications the results of such alignment is not good enough. The usual methodologies to validate the alignment results make use of a tolerance window. If a detected onset falls within the range of the tolerance window it is accepted as correct. If we synthesize the output of such approaches, the time differences within the tolerance window of consecutive onsets lead to a ”non natural” musical performance in terms of tempo regularity. This problematic is shown in 1.1; due to the tolerance window criteria all the instances in the estimation are considered correct but we the output is irregular in time. This irregularity is perceptually unpleasant if we synthesize it.

Thus, applications such as automatic musical accompaniment need a particular alignment that takes care of the regularity of tempo.

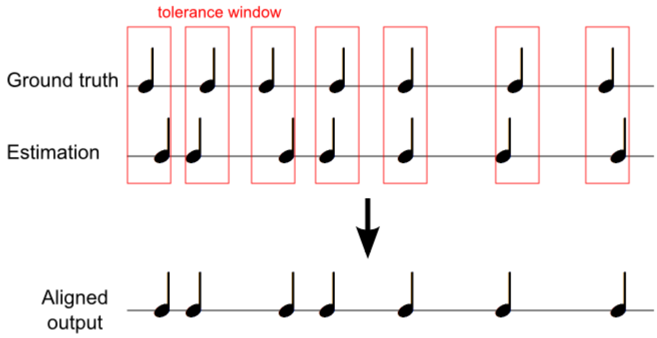


Figure 1.1: Alignment evaluation

1.2 Goals

The main goals of the thesis are the following:

- Provide background and a review of the literature in the field of audio to score alignment
- Develop a new method for online audio to score alignment considering tempo
- Implement different approaches to the problematic
- Evaluate our methods with state of the art techniques in the field.
- Discuss our results, conclude the work and discuss future work and possible applications.

1.3 Structure of the thesis

In chapter two an historical review of score alignment is presented. Furthermore, the related state of the art techniques and approaches to this field are discussed. In chapter three the methodology applied in this research is presented, including the approach to the score alignment process and the feature extaction step as well as the collection of databases and the evaluation methods. In chpater four, results are presented and discussed. Chapter five provides suggestions for future work in the field and summarizes the contributions made in the scope of this research.

Chapter 2

STATE OF THE ART

This section reviews relevant concepts related to audio to score alignment as well as the main used techniques and evaluation methods. It is structured in four subsections. The first one defines audio to score alignment from two different prespectives; online and offline. In the second subsection, the most used techniques nowadays in the field of score alignment are briefly explained. The last section reviews several applications that make use of audio to score alignment. A table that summarize the different approaches to the topic is presented at the end of the section.

2.1 Audio to Score alignment

2.1.1 Overview

Audio to score alignment (or score matching) is the task of synchronizing an audio recording of a musical piece with the corresponding symbolic score. Assuming that a human music performance has tempo variations due to both, musical expression or performance errors, the aim of audio to score alignment is to characterize time differences between the performance and the event timings written in the score. This alignment or synchronization is achieved traditionally by extracting some features from the audio signal and then finding the best match between the feature sequence and the score. A basic diagram of the system is shown in 2.1.

There are two different approaches to the problem depending on how the alignment is processed; offline and online. In the offline version the whole performance is used for the alignment process. This allows the system to ”look into the future” while establishing the matching. This approach is used when the application does not require a realtime alignment process and

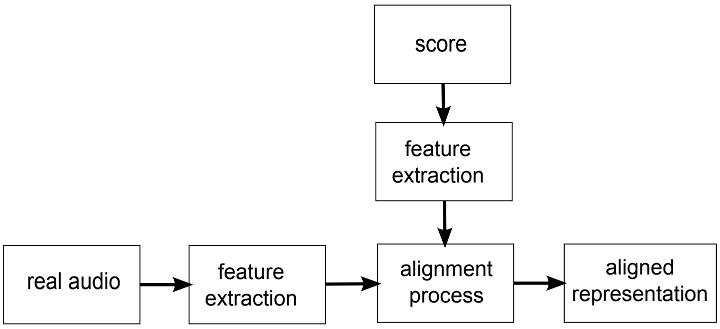


Figure 2.1: Basic audio to score alignment diagram

it can reach higher precision. On the contrary, in the online version (a.k.a Score following) the alignment is performed as the signal is acquired. It can be viewed as the task of deciding "where" is the performance according to its score.

The audio to score alignment problem has been addressed the beginning of the '80 (Dannenberg, 1985; Vercoe, 1984) and in this thirty years we can distinguish two different epochs according to the used techniques. While until 1997 most of the approaches were using string matching techniques, since then two main methods coming from the speech recognition research field were introduced; hidden markov models and dynamic time warping. Nowadays still those are the most common approaches to address score following.

2.1.2 Early approaches to score following

One of the first systems (Dannenberg, 1985) was designed by Dannenberg to follow a monophonic MIDI input using dynamic programming and high-level symbolic representation of the performance. Both the input and the score are converted to strings in order to perform the best match between them. The aim of that system was to perform a real-time musical accompaniment system for soloist musicians. Since this approach was not able to deal with polyphonic signals, as it was designed to deal with monophonic instrument, it was extended later on thinking on piano performances.

In the same year another score following system was presented by Vercoe (Vercoe, 1984) with the same purpose; an automatic accompaniment system. In this "Synthetic Performer" system the author wanted to use pitch as the main audio feature but, because at that time pitch detection was not fast enough to provide reliable results in real-time, information from the finguers of the musician was added. This information was acquired through a series of optical sensors installed on the keys of a flute. Then, the alignment is done by pattern matching techniques. This two primary systems presented slight differences in performance and bigger differences in the approach. While Vercoe's system is more responsive, assuming skilled musicians changing tempo on purpose, Dannenberg achieved more robustness being less confident with the musicians skills or his pitch detection algorithms.

Another approach also based on pitch information was introduced by Puckette in the EXPLODE system (Puckette, 1990; Puckette and Lippe, 1992). The technique used in this score follower is based on a list of previously unmatched notes. A pointer to the current note is used by the algorithm to search for the maximum matching in three steps; first tries to match the played note coming from the performance to a note in the list (to see if it is some skiped note), then tries to match it with the current note and finally to a note in the near future. This system was used for real live music concerts at IRCAM until it was tested with the composition of Philippe Manoury's En Echo in 1993. This piece showed that the use of a finite alphabet of tempered-scale pitches does not work for all the possible music repertoire, leading to a more sophisticated method (Puckette and Jolla, 1995) that take into account the vibrato of the singing voice.

2.2 Currently used techniques for Score alignment

As commented before, in the last two decades mainly two different approaches coming from the research on speech processing have emerged in the field of audio to score alignment. This section describes this two methods and reviews some alternative well-known approaches.

2.2.1 Statistical methods

HMM

Hidden Markov Models (HMMs) are widely used for statistical modeling of nonstationary stochastic processes and are widely used for speech and music. A HMM can be described as a finite state machine where transition between states are ruled by probability functions. In every transition the following state has a value associated to a given probability. Probabilities from one state to another depend only on a limited previous states that is usually set to one. This transition probabilities are modeled as a Markov chain. The states are not directly observable (what is called "hidden") and we can only know the value emitted by each state, that is called observation. A basic diagram is shown in 2.2. The decoding then is the search of the optimal sequence of states given a sequence of observations. The set of parameters of a HMM can be trained to maximize the probability of a given set of observation sequences. A more detailed discussion on HMM theory can be found on Rabiners work.(Rabiner, 1989)

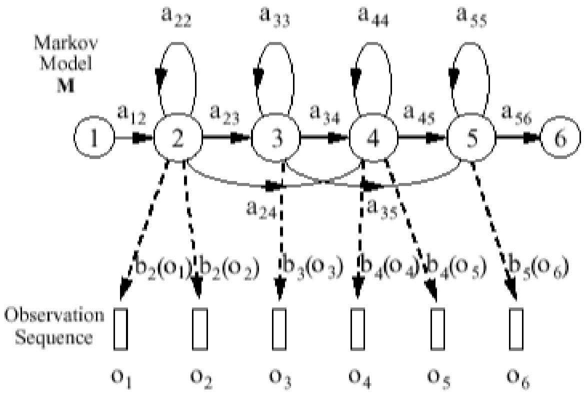


Figure 2.2: Basic diagram of HMM

One of the first approaches to score matching using HMM was introduced in (Cano et al., 1999). Their system is built with three different HMMs models according to each kind of observation; a note, a no-note and a silence. The ”note” model is constructed with three different states that correspond roughly to the energy envelope of a pitched sound; attack, sustain and release. The ”no-note” model, related to non pitched sounds, is built in the same manner. Finally the ”silence” model has a single state due to its nearly flat energy envelope along the time. The lenghts of the notes are modeled with self-transitions and the Viterbi algorithm is performed to achieve the alignment.

Another approach to score following was the system developed in (Raphael, 1999) where a particular decoding technique is used instead of the Viterbi algorithm. According to the information in the score each different kind of note is modeled using several graph topologies due to its time behavoir; long notes, short notes and ornamentations.

A similar approach for the decoding part was used in (Orio and Déchelle, 2001). In this algorithm two categories of states that represent notes are considered; normal states (n-states) and ghost states (g-states). The g-states are used to model three different type of performance errors according to the score; wrong note, extra note and skipped note. Then the performance is modeled by a two level HMM. The sequence of states is then decoded by a dynamic programming based algorithm instead of the classical Viterbi method. This work is detailed and extended in (Cont, 2004).

While most of the previous approaches were designed for monophonic inputs or polyphonic instruments with the same timbre (piano or guitar), in (Cont, 2006) a system to deal with complex polyphonic music was presented. As in the previous algorithms, the score is defined as a hidden Markov chain of states representing sequential events but multipitch observation using Non-negative Matrix Factorization (NMF). The pitch classes used for the matching process are learnt offline using sound examples databases.

Also considering polyphonic music as the input and using the previous works developed in IRCAM for automatic musical accompaniment Raphael developed the Musical Plus One (MPO) (Raphael, 2010). This system is composed of three sub-tasks called "Listen", "Predict" and "Play". In the first step is where the onsets of the notes are identified using the same hidden Markov model approach used in previous works.

Afterwards, a score following system to separate sound sources, the SoundPrism, was presented in (Duan and Pardo, 2011). In this work each frame is represented by a pair of values; score postion and tempo. The sequence of states is inferred using particle filtering.

Other approaches were presented afterwards using also particle filtering for decoding. A particle filter is an algorithm that estimates a latent variable given some observables variables (Arulampalam et al., 2002). It is been used lately for score following algorithms like the one used in (Otsuka et al., 2011). In this approach the observable variable is the audio signal and the latent variables are the score position and tempo. The particle filter estimates the distribution of the position and tempo as the density of particles. Finally, the algorithm outputs three type of values corresponding to the score position, the tempo and an estimation confidence number. According to this estimation value, the system reports either both the score position and tempo or only the tempo, in order to switch between the two levels of the system; "Listen" (just needs tempo) or "Play" (needs both score position and tempo). The auhtors state two main advantages of their approach: enables incremental and simultanious estimation of the score position and tempo and can be done in real-time due to multi-threading computing technique.

2.2.2 Dynamic Time Warping

Overview

Dynamic Time Warping (DTW) is a technique used to align time series or sequences widely used in speech recognition, data mining and information retrieval. The series are presented by 2 vectors $U = u_1, ..., u_n$ and $V = v_1, ..., v_n$ containing feature vectors in each position. The alignment of the two sequences is done by computing the local distances between the different positions. This distances are represented in a $m \times n$ matrix the positions of which are the cost, calculated usually with the Euclidian distance, for aligning each pair (u_i, v_i). A 0 cost means a perfect match and the other costs are all positives. Once this matrix is computed, the DTW algorithm finds the minimum cost path $W = W_1, ..., W_i$. Each W_k represent an ordered pair (i_k, j_k) of aligned vector positions according to the cost matrix built in the previous stage. Then the cost of a path is the sum of the local match costs of the path $D(W)$:

$$D(W) = \sum_{k=1}^{l} d_{U,V}(i_k, j_k)$$

The cost path has to satisfy three constraints to reach the expected results:

1. W is bounded by the end of both sequences: $W_1 = (1, 1)$ and $W_l = (n, m)$
2. W is monotonic: $i_{k+1} >= i_k$ and $j_{k+1} >= j_k$
3. W is continuous

Often other global path constraints are used to reduce the complexity of the computing, i.e. the limitation of the path to lie within a fixed distance of the diagonal.

The minimum cost path is computed by dynamic programming, technique that consists in two stages;

1. Forward step: the lower-cost path is calculated for all the neighbors in the matrix plus the cost to get from the neighbor to the current point. This is done recursively with:

$$D(i,j) = min \left\{ \begin{array}{c} D(i, j - \alpha_j) + d(i,j) \\ D(i - \alpha, j) + d(i,j) \\ D(i - \alpha_i, j - \alpha_j) + \sigma d(i,j) \end{array} \right\}$$

   Where α_i ranges from 1 to I and α_j ranges from 1 to J. To assure diagonality σ is used. Then the cost for more similar frame pairs is zero and the other pairs have positive costs.

2. Traceback stage: the global path is obtained by tracing the recursion backwards from the final position $D(I, J)$.

So alignment of DTW is essentially done in 3 steps:

- Extraction of comparable features from the 2 time series
- Calculation of local distances between the feature vectors of the 2 times series
- Computation of the optimal path with respect to the global distance

A graphic example of the method is shown in 2.3

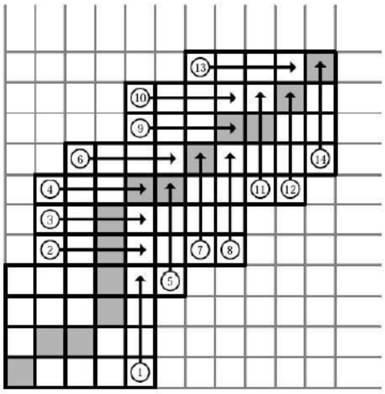


Figure 2.3: Example of dynamic time warping applied to score following presented in (Dixon, 2005)

DTW for score alignment

The DTW algorithm needs two sequences with the same type of data to perform an alignment process. In the case of score following the two sequences are the score and the performace. The score is often treated as an audio signal by converting it to MIDI and then using a synthesizer to create an audio file. (Hu et al., 2003). Once both sequences are audio signals some feature extraction processing is performed.

Most of the approaches take advantage of the spectral representation of the audio data provided by the Fast Fourier Transform (FFT). This representation is often mapped into a more compact version, i.e. the one presented in (Dixon, 2005), where the spectrum is compressed into 84 frequency bins. The frequency axis is logarithmic at high frequencies and linear at low frequencies, providing a simulation of the linear-log frequency sensivity of the human auditory system.

Another mapping of the frequency spectrum is achieved by using the chroma scale (Hu et al., 2003). This vector consists of 12 elements containing the spectral energy envelope of one pitch class. The chroma vector is computed by mapping the frequency bins to nearest step in the chromatic scale. This method proved to perform significantly better than other approaches using MFCC due to the timbre independence of chroma, at is shown in (Hu et al., 2003), and it is still used nowadays for orchestral performance companion (Prockup et al., 2013). A particular use of this feature is presented in (Suzuki et al., 2006). In order to take dynamics into account the sum of chroma and delta chroma vectors is added to the algorithm. First, chroma vectors are normalized by the sum of the elements and second the difference from current and previous frames is computed. This two features characterize not only the pitch classes of the performance but the energy envelope, leading to the best results of the MIREX 2010 Real-time to Score Alignment task.

Several other approaches to the feature extraction process have been presented depending on the particular application and the type of input data. The optimal features for polyphonic music, the alignment of which is more critical according to the selected feature, are discussed in (Joder et al., 2013).

Dixon's Implementation of the DTW algorithm

The first attempt to implement DTW online was done by Simon Dixon (Dixon, 2005). In this approach a low-level spectral representation of the signal is performed as the first step of the processing. Then the data is mapped into 84 frequency bins in a linear-log way; linear for frequencies below 370 Hz and logarithmic with semitone spacing for frequencies from 370 Hz up to 12.5 kHz. The Euclidian distance is used to create a matrix with the distances of the two feature vectors in which the DTW is computed.

The time and space complexity of the standard implementation of DTW is quadratic in the length of the sequences, leading to some limits for an onlie aplication. To solve that, Dixon proposed a constant constraint of DTW in order to get a linear algorithm. Then a forward estimation of the minimum path is performed.

DTW implementation using Spectral factorization

The high accuracy current results in score following (MIREX 2013) is achieved also using a DTW-based algorithm (Carabias et al., 2012). This system has two separate stages; a preprocessing step and the alignment, as it is shown in 2.4. In the first step the different states are defined as an unique combination of notes in a particular time location. These states are used as basis functions for the alignment. A method based on NMF is used to learn the basis functions and then to search these states in each frame of the performance, resulting in a distorsion matrix that shows the cost of each state at each frame. Finally, the alignment is achieved applying the DTW to the distorsion matrix.

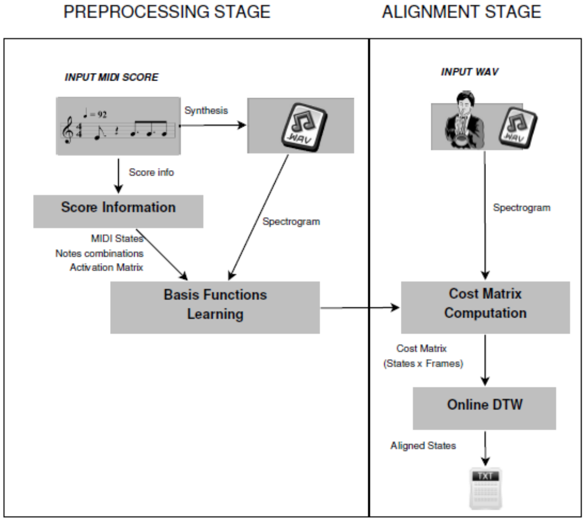


Figure 2.4: Block Diagram of the Proposed Score Follower in (Carabias et al., 2012)

2.2.3 Other approaches

Graphical Models

Graphical models are graphs in which nodes represent random variables. They provide a tool for dealing with uncertainity and complexity, issues that commonly emerge working with real music performances. To deal with such complex systems, graphical models base their design in the notion of modularity, where a complex system is built by combining simpler parts. A detailed discussion is presented in (Murphy, 2001).

C. Raphael was the first to use this approach for audio alignment (Raphael, 2006). He stated that previous approaches were not considering proper models for note lengths; either constraining it to some range or modeled as random, with their distribution depending on a global tempo or learned from past examples. In Raphael's algorithm a note-level model representing both tempo variations and note-by-note deviations is presented. This model is then combined with a model based on pitch information.

Bayesian Networks

Another graphical approach to represent uncertainity is a Bayesian Network (BN). As discussed in (Heckerman, 1995), it is a knowledge representation that combines expert domain knowledge and statistical data. The learning process, similar to the neural networks mehtod, is done by encoding the expert knowledge in a Bayesian network and then using a database to update this knowledge.

A real-time accompaniment system was presented using this method (Raphael, 2001) to represent the joint distribution on the times at which the notes from the soloist and the accompaniment are played. This system consists of three different components; ”listen”, ”synhtesize” and ”anticipate”. The first is modeled using HMM, the second consists is where the audio file is synthesized according to the variables tempo detected in the performance and the third uses the BN to mediate between the two previous states.

This approach was used lately in (Flossmann and Widmer, 2011) for musical retrieval purposes.

2.3 MIREX - Audio to Score Alignment task

The Music Information Retrieval Evaluation eXchange (MIREX)[1] holds a Real-time Audio to Score Alignment task (also known as Score Following) since 2008.

Database

The database consists of 3 datasets composed by recordings of human played peformances and their corresponding symbolic representations of the score (in MIDI format). The audios are recorded with a sample frequency of 44.1 kHz and a 16 bits quantization in wave format. The content of these datasets is:

1. Composed by 46 recordings extracted from 4 distinct musical pieces

[1] http://www.music-ir.org/mirex/wiki/MIREX_HOME

2. Consists of 10 human played J.S. Bach four-part chorales. Each piece is performed by a quartet of instruments: violin, clarinet, tenor saxophone and bassoon that are recorded separately. The recordings are then mixed to create 10 performances with four voices.
3. Composed by 3 piano performances of the Prelude in G minor op.23 - 5 by Serguei Rajmaninov

Evaluation metrics

The main metric to evaluate the Score Following task in MIREX is the aligned rate (AR) or precision. It is defined as the proportion of correctly aligned notes in the score and it ranges from 0 to 1. An note is considered correct if its onset does not deviate more than 2000 ms from the ground truth. The not reported notes that are present in the reference are considered as missed notes, and the notes with start times outside the 2000 ms threshold are considered misaligned notes. Other metrics considered in MIREX are;

- Miss rate: percentage of missed score events
- Misalign rate: percentage of misaligned events
- Mean offset: average sign-valued time offset
- Average offset: average absolute-valued time offset between an estimated note onset and its real onsets.
- Standard offset: standard deviation of sign-valued time offset.
- Average latency: difference between detection time and the time the algorithm processes the audio.

These measures are calculated both over the whole dataset and for each sound, leading to two different precision rates;

- Overall precision rate: percentage of correct aligned score events
- Piecewise precision rate: average for each piece of the value

Last years summary

As commented previously, the Score Following task is been adressed in MIREX since 2008 but it uses the evaluation metrics from the preceding section since 2010. In 2.1 a review of the algorithms presented in the task is shown detailing their alignment technique, the used feature and the total precision.

Year	First author	Features	Alignment technique	Total precision
2013	ChunTa Chen	-	-	67,10 %
2013	Julio J. Carabias	Spectral functions using NMF	DTW	86,70 %
2012	ChunTa Chen	-	-	67,10 %
2012	Ryuichi Sakamoto	chroma, onsets	SCRFs, LDS	52,81 %
2012	Julio J. Carabias	Spectral functions using NMF	DTW	83,01 %
2011	ChunTa Chen	-	-	64,90 %
2011	Kosuke Suzuki	chroma	DTW	67,11 %
2010	Andreas Arzt	low-level spectral representation	DTW	50,84 %
2010	Zhiyao Duan	multipitch, tempo	HMM	49,11 %
2010	Francisco J. Rodriguez Serrano	multipitch	DTW	32,17 %
2010	Francisco J. Rodriguez Serrano	multipitch	DTW	32,44 %
2010	Kosuke Suzuki	chroma and delta chroma	DTW	73,97 %

Table 2.1: Review of MIREX Score Following task

2.4 Applications

This section reviews several applications of audio to score alignment in two separate sections; offline and online applications.

2.4.1 Offline applications

As commented previously, in the offline score alignment the whole performance is available for the alignment process. Thereby, non causal algorithms can be developed, leading to higher matching precision results. In this section the main applications that take advantage of this property are reviewed.

Intelligent audio editors

The widely used audio editing softwares allow multi-track recordings to be manipulated by moving notes, correcting pitch and making other post processes to the audio information. As this work can be costly in time and money due to the increasing demand of the music industry, some ”intelligent” tools were developed to automatically make adjustments to note pitch, timing and dynamic level. This intelligent audio editors, as the one presented in (Dannenberg, 2007), take advantage of the higher accuracies of the offline audio alignment approaches.

Music retrieval applications

One of the challenges in Music Information Retrieval (MIR) is to find the correspondences among different representations of the same music composition. The alignment of these representations is crucial for a proper comparison.

In (Hu et al., 2003) an audio matching and alignment system for music retrieval is presented. The algorithm is based on DTW and several audio features. Both the query and the MIDI database are converted to chroma and MFCC repesentations to obtain two comparable sequences of vectors. After a normalization step, a similarity matrix is create by computing the Euclidian distance between the vectors. The alignment is achieved by computing DTW to the matrix.

A particular approach to MIR is Query-by-humming (QbH). This is basically a music retrieval system that involves taking a user-hummed melody as the input query. The query is then compared to an existing database to obtain a ranked list of music closest to the query. In (Mcnab et al., 1996), string matching techniques are used for such a purpose.

Score-informed Source Separation (Offline)

The approach to source separation that is guided by a musical score is usually known as score-informed source separationand is widely used nowadays. A key ingredient of such a problem is the labeling of audio with symbolic pitches. In order to do that, an audio to score alignment process is imperative.

One of the first examples of this method is presented in (Woodruff et al., 2006). In this approach a source separation system is implemented aiming to separate sound sources from stereo mixtures to allow remixings of the recordings. The system uses knowledge of the written score and spatial information from an anechoic, stereo mixture. Here, the alignment is performed by converting the MIDI file (score) and the audio file into a chromagram representation. Then the next step uses DTW to find the best alignment using the Euclidian distance between the two chroma vectors.

A more recent approach to source separation using knowledge from the score is presentd in (Hennequin et al., 2011). In this case the information in the score is used to initialize an algorithm which computes a representation of the spectrogram. This representation is computed with a non-matrix factorization (NMF) technique. The separation of the sources in the mixture is achieved with time-frequency masks that the algotihm provides.

Another approach, using also NMF, is the one related in (Fritsch, 2013). Here, the separation method is composed of two different phases consisiting of two consecutive NMF routines; one to learn the components of each insturment from the score and the other to fit these components to the actual mixture. As in many other cases, a MIDI score is used to synthesize each instrument separately. Then, a dynamic time warping (DTW) algorithm is used to align the synthesized signals on the mix.

A review of the research problem and its state of the art techniques is presented in (Ewert et al., 2014).

2.4.2 Online applications

The online approach to score alignment, usually known as score following, has a broad variety of applications. These applications are real-time so the alignment algorithm has to deal with the trade off between accuracy and velocity.

Automatic musical accompaniment

The first automatic musical accompaniment of a soloist musician was developed by Dannenberg (Dannenberg, 1985) using string matching techniques and dynamic programming. In this approach, the computer is given a score containing parts for the soloist and for the corresponding accompaniment. The problem is then divided in three different parts;

- recognise what the soloist is doing
- match the input to the score
- produce and accompaniment

For the alignment process first the score and the audio input are converted to strings and then the best match between these strings is computed. A basic diagram of the system is shown in 2.5.

Another more recent approach to automatic musical accompaniment is the Antescofo system presented by Cont(Cont, 2007). The system is able to anticipate the events and to respond to them in real-time while reproducing electronic scores. It is based in an statistical approach to score following but it enables, in advanced use, temporal interaction between the performance and the electronic score.

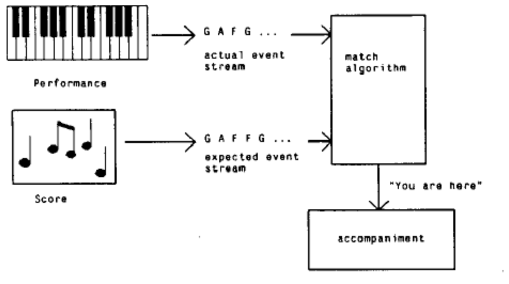


Figure 2.5: Diagram of the musical accompaniment shown in (Dannenberg, 1985)

Afterwards, several automatic musical accompaniment systems considering different musical inputs were developed by the same author. Some examples are the approach focused in musical ensembles (Grubb and Dannenberg, 1994) and using piano(Dannenberg and Raphael, 2006).

Automatic page turning

One of the most succesful applications of score following is the Polyphonic Score Following (PSD) algorithm used in the Tonara Ltd. system. This approach is based on stochastic alignment to track the musician's position in the score regardless of performance mistakes, noise or tempo variations. An automatic page-turning based on Dixon's work (Dixon, 2005) is presented in (Arzt, 2007). The score is represented here as MIDI and then converted into an audio file by a software synthesizer. Then the matcher step receives one audio frame from the performance, calculates its feature vector, hands it over to the matching algorithms, waits until they are done and then start with the next frame. The architecture of the system is shown in 2.6.

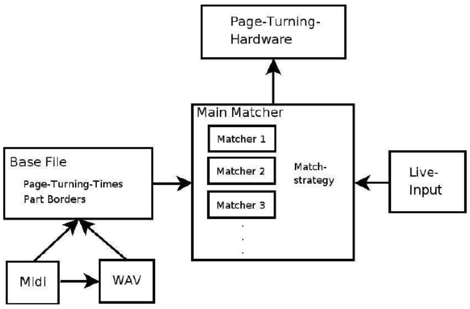


Figure 2.6: Architecture of the Automatic Page-Turner from (Arzt, 2007)

Score-informed Source Separation (Online)

The first online system that addressed score-informed source separation in an online fashion is the SoundPrism system presented by Duan in (Duan and Pardo, 2011). The aim of system is to separate single-channel polyphonic music into source signals. As commented before, it is essential to have the performance aligned to the score before performing the source separation step.

An HMM approach is used to model the score follower step. The observation model is based on multi-pitch estimation and the score position and tempo are inferred using particle filtering, as commented previously.

First author	Techniques	Features	Year
B. Vercoe	String matching	Tempo and pitch	1984
R. Dannenberg	String maching	Pitch	1984
B. Vercoe	String maching	Pitch	1990
L. Grubb	DP, string matching	Pitch	1994
M. Puckette	DP, string matching	Instantaneous pitch	1995
R. Mcnab	String maching	Pitch, onsets	1996
P. Cano	HMM	F0, F0 error, Delta F0, ZCR, energy, Delta energy	1999
C. Raphael	HMM	Pitch	1999
N. Orio	DTW	Spectral peaks plus a model of attacks and silences	2001
N. Orio	HMM	Energy envelope, log energy, even-aodd harmonics ratio	2001
C. Raphael	Bayesian Network, HMM	Note onset times	2001
N. Hu	DTW	Chroma, MFCC	2003
S. Dixon	DTW modified for online purposes	Spectrum	2005
K. Suzuki	DTW, locally constrained	Sum of chroma and delta chroma vectors	2006
C. Raphael	Hybrid graphical model	Tempo and pitch	2006
A. Cont	Hierarchical HMM, NMF	Multi-pitch	2006
R. Dannenberg	DTW	RMS and F0	2007
A. Artz	DTW (Dixon approach)	Spectrum	2007
C. Raphael	Bayesian Network, HMM	Tempo and pitch	2009
C. Raphael	HMM	Note onset times	2010
T. Otsuka	Particle Filters	Tempo and pitch	2011
Tonara Ltd.	Stochastic approach	unknown	2011
Z. Duan	HMM, particle filters	Multi-pitch, tempo	2011
J.J. Carabias	DTW, Spectral factorization	Spectral templates using NMF	2012
M. Prockup	DTW	Chroma	2013

Table 2.2: Table with an historical review of audio to score alignment systems

Chapter 3

METHODOLOGY

In this section, we describe the proposed framework for realtime audio-to-score alignment. In 3.1 the block diagram of the proposed method is shown. As can be seen, the framework has two stages. The preprocessing stage must be computed beforehand and only the MIDI score is required. Then, once the parameters are learned, alignment can be computed in realtime.

The successive stages displayed in 3.1 are detailed below.

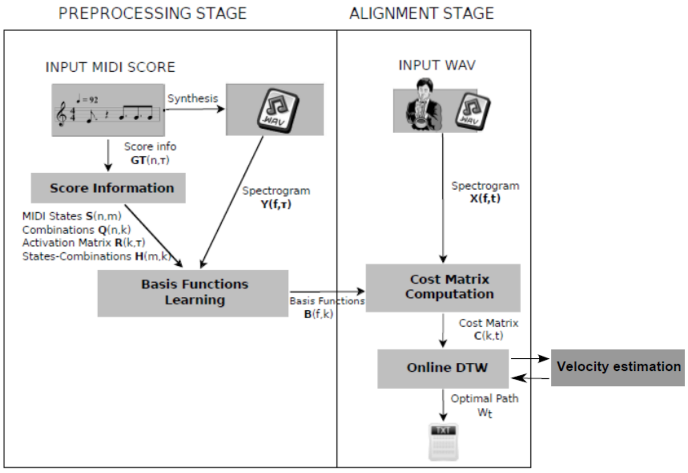


Figure 3.1: Block diagram of the proposed method

3.1 Preprocessing Stage

The pre-processing stage is taken from the approach by J.Carabias (Carabias et al., 2012) that performed the best result in the MIREX 2012 Score Following task. In this stage, the parameters for the alignment are learned from the score, which must be provided beforehand using MIDI representation. This stage is performed in two successive steps; states definition and spectral patterns learning, that are detailed below.

3.1.1 States Definition

The aim of this stage is to compute a compressed representation of the score. First, each unique combination of notes is computed. Then, the states define the sequence of this notes combinations, as is shown in 3.2. To compute this states, first the binary ground-truth transcription matrix $\mathbf{GT}(n, \tau)$ (see figure 3.2(a)) is inferred from the MIDI score, where τ is the time in frames referenced to the score (MIDI time) and n are the notes in MIDI scale. In figure 3.2(a) the MIDI score involves just one instrument (a piano) but more instruments can be defined in a score. for that cases n index refers to each note of the different instruments. Consequently, the number of total notes for a composition, N, is obtained as the number of different notes per instrument multiplied by the number of different instruments. The score defines a consecutive sequence of M states. Each state m is defined by its combination of notes (for all instruments). Also, the score informs about the time changes from one state to the next state. In fact, a score follower must determine the time (referenced to the input signal) of all transitions between states. There are only K unique combination of notes in a score where $K \leq M$ because some states represent the same combination of notes.

From the ground-truth transcription matrix $\mathbf{GT}(n, \tau)$, we obtain the following decomposition of binary matrixes:

$$\mathbf{GT}(n, \tau) = \mathbf{Q}(n, k)\mathbf{R}(k, \tau) \tag{3.1}$$

where $\mathbf{Q}(n, k)$ is the notes-to-combination matrix, k the index of each unique combination of notes, K the number of unique combinations for the score and $\mathbf{R}(k, \tau)$ represents the activation of each combination in MIDI time. In figure 3.2(c), the note-to-combination matrix $\mathbf{Q}(n, k)$ is represented. This matrix contains the notes belonging to each combination but no information about MIDI time. Conversely, $\mathbf{R}(k, \tau)$ matrix retains the MIDI time activation per combination but no information about the notes active per combination, as can be seen in figure 3.2(b).

In order to obtain the information for MIDI states required to perform the alignment, the notes-to-combination matrix $\mathbf{Q}(n, k)$ is further decomposed as

$$\mathbf{Q}(n, k) = \mathbf{S}(n, m)\mathbf{H}(m, k) \tag{3.2}$$

where $\mathbf{S}(n,m)$ is the notes-to-state matrix, m the index for the MIDI states, M the number of states and $\mathbf{H}(m,k)$ represents the unique combination k of notes active at each state m. In figure 3.2(e), the notes-to-state matrix $\mathbf{S}(n,m)$ is represented, this matrix contains the notes belonging to each state, while $\mathbf{H}(m,k)$ matrix informs about the combinations active at each state, as can be seen in figure 3.2(d).

The matrixes here defined will be used in the next stages to perform the alignment and are computed from the MIDI score.

3.1.2 Spectral Patterns Learning

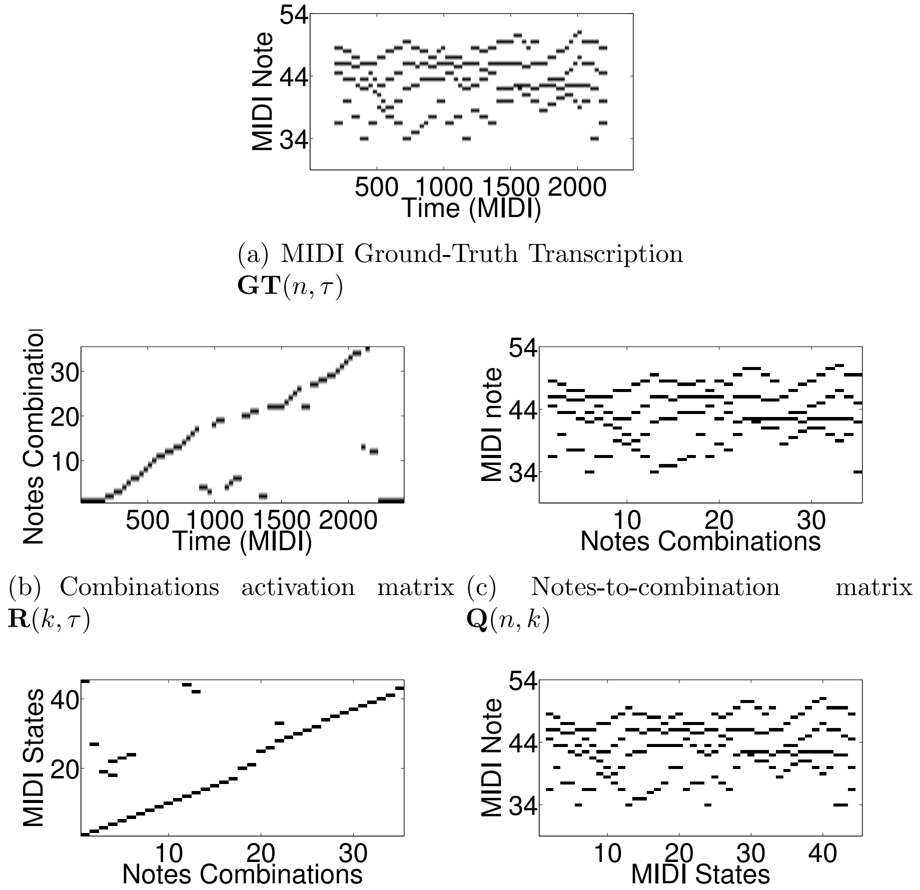
When a signal frame is given to a score follower, the first step should be to compute a similarity measure between the current frame and the different combinations of notes defined by the score. Our approach is to compute a distortion between the frequency transform of the input and just one spectral pattern per combination of notes. A spectral pattern is here defined as a fixed spectrum which is learned from a signal with certain characteristics. The use of only one spectral pattern per combination allows us to compute the distortions with a low complexity signal decomposition method. This means that our method must learn in advance the spectral pattern associated to each unique combination of notes for the score. To this end, a state-of-the-art supervised method based on Non-Negative Matrix Factorizacion (NMF) with Beta-divergence and Multiplicative Update (MU) rules is used, but in this work, we propose to apply it on synthetic signal generated from the MIDI score instead of the real audio performance.

First of all, let us define the signal model as

$$\mathbf{Y}(f,\tau) \approx \hat{\mathbf{Y}}(f,\tau) = \mathbf{B}(f,k)\mathbf{G}(k,\tau) \tag{3.3}$$

where $\mathbf{Y}(f,\tau)$ is the magnitude spectrogram of the synthetic signal, $\hat{\mathbf{Y}}(f,\tau)$ is the estimated spectrogram, $\mathbf{G}(k,\tau)$ matrix represents the gain of the spectral pattern for combination k at frame τ, and $\mathbf{B}(f,k)$ matrix, for $k = 1,...,K$, represents the spectral patterns for all the combinations of notes defined in the score.

When the parameters are restricted to be non-negative, as it is the case of magnitude spectra, a common way to compute the factorization is to minimize the reconstruction error between the observed spectrogram and the modeled one. The most popular cost functions are the Euclidean (EUC) distance, the generalized Kullback-Leibner (KL) and the Itakura-Saito (IS) divergences. Besides, the Beta-divergence (see eq. 3.4) is another commonly



(a) MIDI Ground-Truth Transcription $\mathbf{GT}(n, \tau)$

(b) Combinations activation matrix $\mathbf{R}(k, \tau)$

(c) Notes-to-combination matrix $\mathbf{Q}(n, k)$

(d) States-to-combination matrix $\mathbf{H}(m, k)$

(e) Notes-to-state matrix $\mathbf{S}(n, m)$

Figure 3.2: (a) MIDI Ground-Truth Transcription. (b) Combinations activation matrix. (c) Notes-to-combination matrix (d) States-to-combination matrix (e) Notes-to-state matrix

used cost function that includes in its definition the three previously mentioned EUC ($\beta = 2$), KL ($\beta = 1$) and IS ($\beta = 0$) cost functions.

$$D_\beta(x|\hat{x}) = \begin{cases} \frac{1}{\beta(\beta-1)}\left(x^\beta + (\beta-1)\hat{x}^\beta - \beta x \hat{x}^{\beta-1}\right) & \beta \in (0,1) \cup (1,2] \\ x \log \frac{x}{\hat{x}} - x + \hat{x} & \beta = 1 \\ \frac{x}{\hat{x}} + \log \frac{x}{\hat{x}} - 1 & \beta = 0 \end{cases} \tag{3.4}$$

In order to obtain the model parameters that minimize the cost function, in (Lee et al., 2000) Lee proposes an iterative algorithm based on MU rules. Under these rules, $D_\beta(\mathbf{Y}(f,\tau)|\hat{\mathbf{Y}}(f,\tau))$ is shown to be non-increasing at each iteration while ensuring non-negativity of the bases and the gains. Details are omitted to keep the presentation compact, for further information please read (Lee et al., 2000; Nantes, 2011). For the model of eq. (3.3), multiplicative updates which minimize the Beta-divergence are defined as

$$\mathbf{B}(f,k) \leftarrow \mathbf{B}(f,k) \odot \frac{\left(\mathbf{Y}(f,\tau) \odot \hat{\mathbf{Y}}^{\beta-2}(f,\tau)\right)\mathbf{G}^T(\tau,k)}{\left(\hat{\mathbf{Y}}^{\beta-1}(f,\tau)\right)\mathbf{G}^T(\tau,k)} \tag{3.5}$$

$$\mathbf{G}(k,\tau) \leftarrow \mathbf{G}(k,\tau) \odot \frac{\mathbf{B}(f,k)\left(\mathbf{Y}(f,\tau) \odot \hat{\mathbf{Y}}^{\beta-2}(f,\tau)\right)}{\mathbf{B}(f,k)\left(\hat{\mathbf{Y}}^{\beta-1}(f,\tau)\right)} \tag{3.6}$$

where operator $\odot$ indicates Hadamard product (or element-wise multiplication), division and power are also element-wise operators and $(\cdot)^T$ denotes matrix transposition.

Finally, the method to learn the spectral patterns for each state is described in Algorithm 1.

Algorithm 1 Method for learning spectral patterns combinations

```
1 Initialize $\mathbf{G}(k,\tau)$ as the combinations activation matrix $\mathbf{R}(k,\tau)$ and
  $\mathbf{B}(f,k)$ with random positive values.
2 Update the bases using eq. (3.5).
3 Update the gains using eq. (3.6).
4 Normalize each spectral pattern of $\mathbf{B}(f,k)$ to the unit $\beta$-norm.
5 Repeat step 2 until the algorithm converges (or maximum number of
  iterations is reached).
```

As explained in section III-A1, $\mathbf{R}(k,\tau)$ is a binary combination/time matrix that represents the activation of combination k at frame τ of the training data. Therefore, at each frame, the active combination k is set to one and the rest are zero. Gains initialized to zero will remain zero, and therefore the frame becomes represented with the correct combination.

The case of the combination in which there is not any active notes must be initialized beforehand. In those MIDI frames where no signal is defined (silence frames) and in order to avoid numerical problems, we propose to initialize the spectral patterns associates to silence combination to a constant. Note that a similar solution has been used in (Gemmeke et al., 2011) to overcome the silence problem on an automatic speech recognition framework.

3.2 Alignment

In this stage, the alignment between the score and the audio performance is accomplished in realtime once computed the information from the preprocessing stage.

3.2.1 Observation Model

As explained in section 3.1.2, the spectral patterns $\mathbf{B}(f,k)$ for the K different combinations of notes are learned in advance using a MIDI synthesizer and kept fixed. Each spectral pattern models the spectrum of a unique combination.

Now, the aim is to compute the gain matrix $\mathbf{G}(k,\tau)$ and the cost matrix $\mathbf{D}(\tau,t)$ that measures the suitability of each combination of notes belonging to each MIDI time τ to be active at each frame t (referenced to the signal input) by analyzing the likelihood between the spectral patterns $\mathbf{B}(f,k)$ and the input signal spectrogram[1]. From the cost matrix $\mathbf{D}(\tau,t)$, a classical DTW approach can be applied to compute the alignment path.

To this end, we propose to use the realtime single-pitch constrained method proposed in (Carabias-Orti et al., 2013). Although this method was designed to address music transcription of monophonic signals, it can be adapted for Score Following of polyphonic signals because only one combination will be active at a time. In this transcription method, the optimum combination k_{opt} is chosen to minimize the Beta-divergence function at frame t under the assumption that only one gain is non-zero at each frame. Taking the combinations as the index of gains $\mathbf{G}(k,\tau)$, this assumption is fair because only a unique combination k of notes is active at each time (at least when producing the audio signal).

[1] Note that we are using $\mathbf{X}$ and t instead of $\mathbf{Y}$ and τ to represent the signal magnitude spectrogram and the time frames to distinguish between real world and synthetic signals.

Thus, the signal model with the single-combination constraint for the signal input vector at time t, $\mathbf{x}_t(f)$, is defined as follows.

$$\mathbf{x}_t(f) \approx \hat{\mathbf{x}}_{k_{opt},t}(f) = g_{k_{opt},t}\mathbf{b}_{k_{opt}}(f) \tag{3.7}$$

where $\hat{\mathbf{x}}_{k_{opt},t}(f)$ is the modeled signal for the optimum combination k_{opt} at frame t.

$$k_{opt}(t) = \arg\min_{k=1,\ldots,K} D_\beta\left(\mathbf{x}_t(f)|g_{k,t}\mathbf{b}_k(f)\right) \tag{3.8}$$

The signal model assumes that when combination k is active all other combinations are inactive and, therefore, the gain $g_{k,t}$ is just a scalar and represents the gain of the k combination. The model of eq. (3.8) allows the gains to be directly computed from the input data $\mathbf{X}(f,t)$ and the trained spectral patterns $\mathbf{B}(f,k)$ without the need of an iterative algorithm, making the computation really fast. To obtain the optimum combination at each frame, we must first compute the distortion obtained by the projection of each combination at each frame and then select the combination that achieves the minimum distortion as the optimum combination at each frame.

For Beta-divergence, the cost function for combination k and frame t can be formulated as

$$D_\beta(\mathbf{x}_t(f)|g_{k,t}\mathbf{b}_k(f)) = \\ \sum_f \frac{1}{\beta(\beta-1)}\left(\mathbf{x}_t^\beta(f) + (\beta-1)(g_{k,t}\mathbf{b}_k(f))^\beta - \beta\mathbf{x}_t(f)(g_{k,t}\mathbf{b}_k(f))^{\beta-1}\right) \tag{3.9}$$

The value of the gain for combination k and frame t is then computed by minimizing eq. (3.9). Conveniently, this minimization has a unique non-zero solution due to the scalar nature of the gain for combination k and frame t (see more details in (Carabias-Orti et al., 2013)).

$$g_{k,t} = \frac{\sum_f \mathbf{x}_t(f)\mathbf{b}_k(f)^{(\beta-1)}}{\sum_f \mathbf{b}_k(f)^\beta} \tag{3.10}$$

Finally, the distortion matrix for each combination at each frame is defined as:

$$\mathbf{\Phi}(k,t) = D_\beta(\mathbf{x}_t(f)|g_{k,t}\mathbf{b}_k(f)) \tag{3.11}$$

where β can take values in the range $\in [0,2]$.

As can be inferred, the distortion matrix $\mathbf{\Phi}(k,t)$ provides us information about the similitude of each combination k spectral pattern with the real signal spectrum at each frame t. Using this information, we can directly compute the cost matrix between the MIDI time τ and the time of the input signal t as

$$\mathbf{D}(\tau, t) = \mathbf{R}^T(\tau, k)\mathbf{\Phi}(k, t) \tag{3.12}$$

where $\mathbf{R}(k, \tau)$ is the combinations activation matrix defined in section III-A1. The process is detailed in Algorithm 2.

Algorithm 2 Distortion matrix computation method

```
Initialize $\mathbf{B}(f, k)$ with the values learned in section 3.1.2.
for t=1 to T do
  for k=1 to K do
    Compute the gains $g_{k,t}$ using eq. (3.10).
    Compute current value the distortion matrix $\mathbf{\Phi}(k, t)$ using eq. (3.11).
  end for
end for
Compute the cost matrix $\mathbf{D}(\tau, t)$ between MIDI time and input signal time using (3.12).
```

3.2.2 Path Computation

The computation of the maximum similarity path part is the main concern of this master thesis. We chose a Dynamic Time Warping (DTW) approach due to its good results in the MIREX Score Following task, a comparison of the algorithms presented for this task in the latest years is presented in 3.1. As it is shown, the DTW performs better than other approaches for this task.

Year	DTW-based algorithms	Other algorithms	Best approach (total precision)
2013	1	1	DTW (86,70 %)
2012	1	2	DTW (83,01 %)
2011	1	1	DTW (67,11 %)
2010	4	1	DTW (73,97 %)

Table 3.1: Comparison of MIREX Score following task algorithms

3.2.3 Offline DTW based on Ellis

The beginnning of this master thesis implementation of DTW is the one provided by Dan Ellis [2] in a DTW Toolbox for MATLAB. The provide routines are:

- simmx.m: a utility to calculate the full local-match matrix i.e. calculating the distance between every pair of frames from the sample and template signals.
- dp.m: implementation of the simple dynamic programming algorithm that allows three steps - (1,1), (0,1) and (1,0) - with equal weights.
- dp2.m: experimental alternative version that allows 5 steps - (1,1), (0,1), (1,0), (1,2), and (2,1) - with different weights to prefer sloping paths but without a hard limit on regions in which matches are found.
- dpfast.m: fast version that uses a MEX routine (dpcore) to execute the non-vectorizable inner loop. Also allows user-specified step/cost matrix.
- dpcore.c: C source for the MEX routine that speeds up dpfast.m.

We make use of the dpfast.m and dpcore.m to built our first offline approach DTWofflineTmxTr. From the distorsion martrix coming from NMF process a matrix of real time against midi time is built, simply by replicating the states according to its durations (taken from the midi score). The final similarity matrix on which the DTW is applied is shown in 3.3, using the sample 01-ba-cl-sx-vl from the polyphonic database provided by prof. Duan [3]. The similarity values are shown with the color scale used in MATLAB for the imagesc function; the blue range colors stand for high similarity and the red range colors for low. A maximum similarity path is clearly displayed around the diagonal of the matrix. In order to mantain the diagonality of the estimation we "tune" the cost matrix. We restrict the costs proposed by Ellis, that allows three different type of steps (1,1), (0,1) and (1,0), to only two steps (1,1) and (1,0), considering that a group of notes in the real audio can only be estimated as one state in the score. This leads the algorithm to always move forward. Both the similarity matrix and the cost matrix are passed to the Ellis code, that first computes the cumulated cost matrix and finally gives us the global minimum path. The path is achieved thanks to

[2] D.Ellis `http://www.ee.columbia.edu/~dpwe/resources/matlab/dtw/`

[3] Duan `http://www.ece.rochester.edu/~zduan/resource/Resources.html`

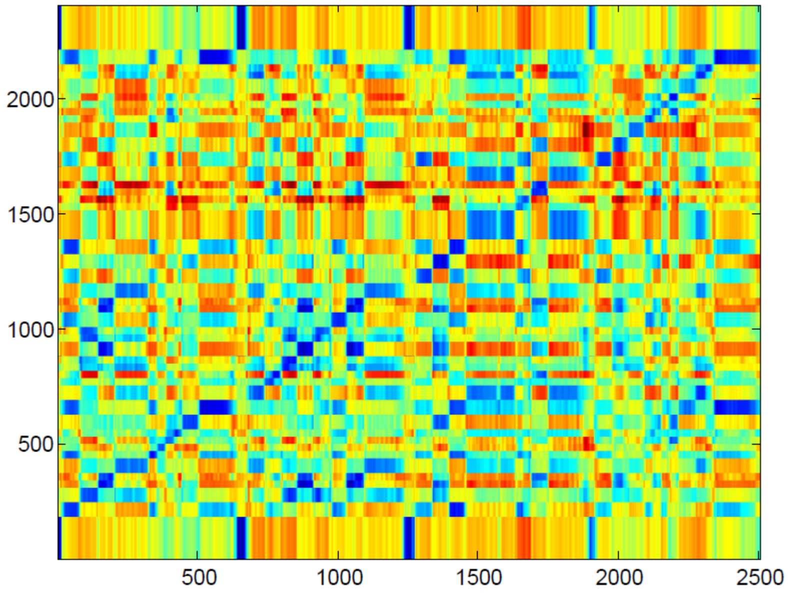


Figure 3.3: Similarity matrix

a traceback algorithm performed from the antidiagonal of the matrix. An example of the global minimum path estimated with this approach is shown in 3.4, using the same sample.

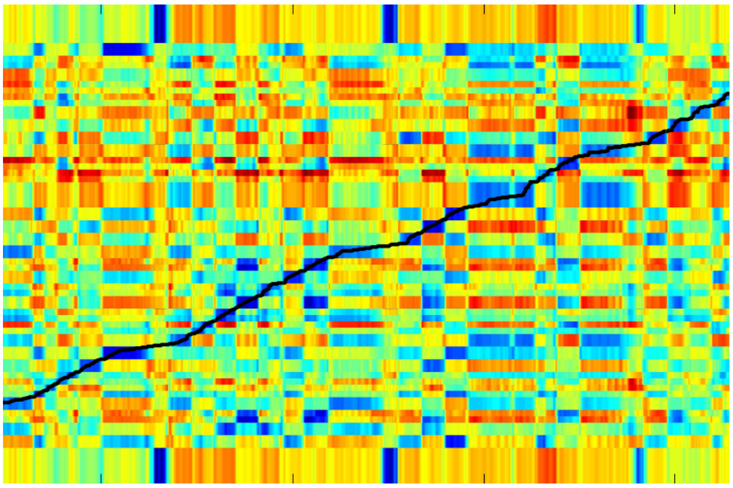


Figure 3.4: Global minimum path calculated by offline approach on top of the similarity matrix

As its shown, this approach is highly effective in terms of finding the correct path and could be used for aplications that do not need real-time alignment.

3.2.4 Online implementation

As one of the goals of this thesis is the implementation of an alignment algorithm for an automatic accompaniment system, it own nature leads to an online approach to the alignment process. For such a purpose we implemented an online version of the previous approach DTWonlineTmxTr. In a straightforward fashion, this approach gives an estimated position in the midi score for each time. The DTW is calculated only for the current time and then the estimated position is given as the minimum cumulated cost of the time. A result of this approach is shown in 3.5. As some notes are played more than one time in the performance and some other are not played in the way expected by the score, their feature vectors are more similar to other feature vectors in different places of the score, leading to the wrong estimations of the path shown in the figure. Although the algorithm is given the maximum similarity for each time, it does not have musical meaning because a score is played usually in a monotonic fashion; one event consecutively to the previous one. In order to achieve a variation of the path around the diagonal of the matrix, meaning that the performance is following the events of the score consecutively, we propose several restrictions for the minimum search in the next section.

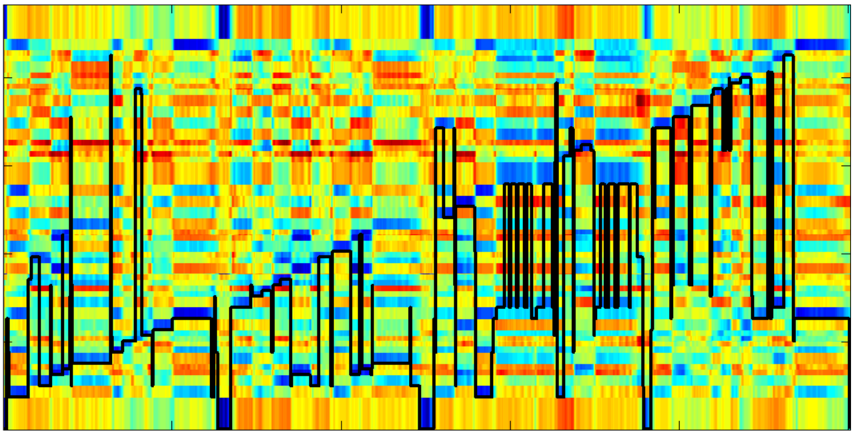


Figure 3.5: Global minimum path calculated by online approach on top of the similarity matrix

3.2.5 Online DTW with restrictions

In order to restrict the minimum path search in some particular time ranges, we first propose a global restriction according to our musical background; a score is usually played with tempo variations no larger than 4 times slowly and 4 times faster. Assuming that the tempos for music peformance are usually between 40 bpm and 240 bpm, playing 4 times faster or slower will still stand in the usual tempo range. To achieve that, the search of the minimum is restricted in three different ways for each state/group of notes. The state duration is defined as the number of frames of the state in the original score.

1. time is below $stateduration/4$: the search is restricted to the duration of the state

2. time is between $stateduration/4$ and $stateduration * 4$: the search is restricted to the current and next state

3. time is above $stateduration * 4$: the search is restricted to the next state.

To allow this type of jumps the cost matrix proposed by Ellis is also changed to consider (1,2),(1,3),(1,4),(2,1),(3,1) and (4,1) steps. With this restrictions the algorithm only searchs for local similarities and we avoid the radical jumps produced in the previous approach due to similarities among different times in the score. Even so, the algorithm tends to change a lot due to near similarities and, although in a general point of view it keeps close to the diagonal, is not as regular as the tempo performance.

3.2.6 DTW with estimation velocity

In real musical performances the tempo is usually changed regularly and stretched around a basis tempo. For this novel approach we assume that the note onset that we want to estimate is probably related to the tempo the performer is been playing up to that time. With this assumption we built a model that compute the estimation velocity and change the cost matrix according to it in order to push the DTW in the desired direction. The velocity is calculated as the mean of the type of jump that the path does in a particular range time. The mean was chosen for its better results after testing with different statistical operators as meddian or mode. The range time is the duration of 4 consecutives states and it was chosen empirically after some tests. The weights of the cost matrix are changed according to the calculated estimation velocity; the weight of the step more similar to the velocity is kept as 1, the weigths of the closest steps are changed to 1.1 and the others to 1.3. An example is presented in 3.6.

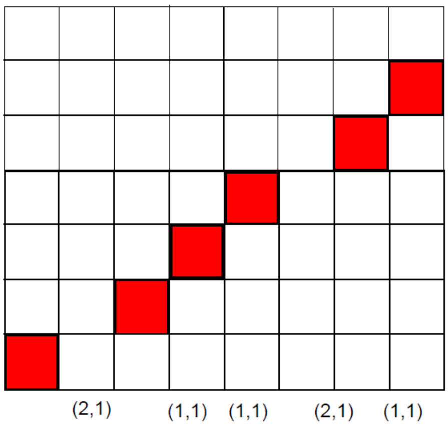


Figure 3.6: Example of jumps type

The estimation velocity is computed as $mean(typeofjump) = (3 * 1 + 2 * 2)/5 = 1,4$ so we would kept (1,1) = 1, change (1,2) = 1,1 and change the others to 1,3.

The DTWonlinerestrictions algorithm uses the global constraints from the previous section and the DTW tunning proposed here. An estimated score position for each time is shown in 3.7.

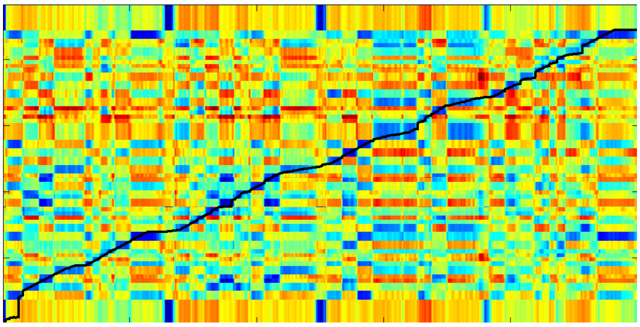


Figure 3.7: Global minimum path calculated by online approach with restrictions on top of the similarity matrix

3.2.7 Anchor points decision

It is been shown that, although the local restrictions of DTW help the regularity of the estimated path, there are some places in the score where the algorithm has not the certainty of where to go. Moreover, there are other places where the certainity of which state is being played is so high. This certainty happens in the states with notes that only appear few time in the score. Computing the spectrum cross-correlation of such state gives substantial lower results than computing it for a state with notes that appear more often in the score. These particular places in the score are taken as "anchor" points where the estimation velocity is calculated. Before starting the alignment process, a correlation between each state and the rest is calculated. This correlation is then summed to obtain the total similarity between each state and the others. Using the findpeaks MATLAB function to the -1 matrix obtained in the previous step, several minimum point are selected. We restrict the minimum peak distance to 3 to avoid having too close anchor points. An example of the summed correaltion is shown in 3.8. If some of the selected states is a silence it is discarded due to the fact that silence can be used by musicians to change tempo.

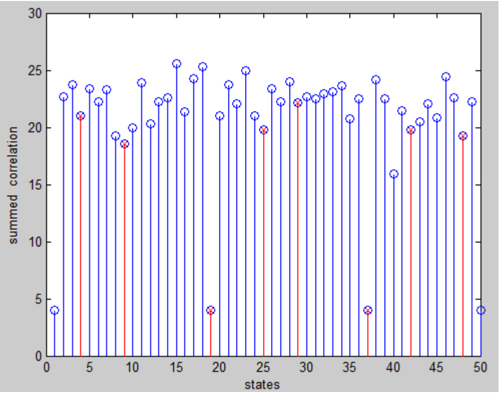


Figure 3.8: Summed correlation between each state and the others with the "anchor" states in red

3.3 Evaluation

3.3.1 Introduction

In this section the evaluation criteria is reviewed, as well as the used music collections. The evaluation is splitted into two categories; one from a quantitative point of view based in the MIREX evaluation framework (Davies et al., 2009) and the other from a qualitative point of view related to perceptual experimenst. As one of the goals of this thesis is to implement an algorithm to be used by an automatic accompaniment system, we want to provide a perceptual improvement of the aligned audio in the sense that some tempo regularity is kept in the output of the system. This goal confronts directly with the pursued goal of having a high precision but we want to achieve a trade-off between ”naturality” and accuracy.

3.3.2 Music Collections

Although larger digital music databases can be found nowadays, for our particular task we need both the transcripted MIDI score and the manually time-aligned ground truth file. In this section we present one well known music collection widely used to validate score alignment algorithms. We also used the MIREX database to test our final algorithm but, since the ground truth is not provided in this collection, the results are not reviewed in this section.

Bach10 Dataset

In order to adress some music processing research problems, Duan and Pardo (Duan and Pardo, 2011) prepared the Bach10 Music Dataset. It is a polyphonic database that can be used for Audio-score Alignment as well as Multipitch Estimation tracking and Source Separation. The dataset consists of of the audio recordings of each part and the ensemble of ten pieces of four-part J.S.Bach chorales, as well as their MIDI scores, the annotated ground-truth for each piece. The instruments of the audio recordings are violin, clarinet, saxophone and bassoon.

The three different types of data were generated diferently; while the MIDI files were downloaded from the Internet the audio recordings were recorded part by part with the musicians isolated while listening the recordings of others through headphones.

The ground-truth pitches and notes of the audio recordings were first generated for each part and then combined with other parts. The audio of the mix and each individual part were processed with a window size of 46 ms and a 10 ms hop. The first window was centered at 23 ms from the beginning. For each frame of the mix a RMS threshold of 0.075 is applied to discard the unvoiced ones. For theses frames no ground pitch of any individual part is detected. In the voiced frames a robust single pitch detection algorithm (de Cheveigne and Kawahara, 2002) is performed for each individual part to detect the ground pitch value. Then some manual corrections were done to fix some apparent errors. Ground-truth notes were then formed by connecting ground-truth pitches in adjacent frames manually.

The alignment between audio and MIDI used as ground-truth were obtained through human annotation. The auhtors built a software to record and modify human tapped beats. A musician tapped beats using a keyboard while listening to the adio file. This way the obtained a ground truth alignment between audio beat times and MIDI beat times. Then, in the text file the beat time alignment for each note in the MIDI file is linearly interpolated.

The data of this music collection it also can be extended by exploring the combinations of different parts of each piece. The maximum number of audio recordings that can be generated for each piece is 15, containing four monophonic parts, six duets, four trios and one quartet. Although the temporal dynamics of these new recordings are the same as the original one, they can be used to test algorithms in different polyphonies and instrumentations.

3.3.3 MIREX measures

We have used the same evaluation metrics than in MIREX, as commented in 2.3. However, we decided to use an evaluation window of 200 ms instead of the 2000 ms window used in the MIREX Score Following task. We have considered that a window this size is too large to distinct the tiny changes in time that a musician can do while playing..

3.3.4 Subjective evaluation

As one of the goals of this thesis is to increase the regularity of the alignment process a perceptual evaluation shold be done to test if this goal is achieved. We want the algorithm to produce an output that, once synthesized, could provide a sense of "naturality" for the audience. For such a purpose we synthesize the outputs of the different approaches to design a straightforward perceptual test. The synthetic signals are generated using Timidity++ with the FluidR3 GM soundfont in the same way that in the pre-process stage of the algorithm. The test should be done by musicians, musicologists or users with high musical knowledge. The synthesized audios will be presented to the users blindly (without knowing wich algorithm is used). The users will be asked to rate the sounds according to how natural they sound to them.

Chapter 4

RESULTS

4.1 Introduction

To evaluate our hypothesis we implemented four different algorithms according to the different steps explained in chapter 3. We tested all the algorithms with the music collection discussed in 3.3.2 in order to get comparable results of our approaches to the problem. We have been using the following algorithms;

- Offline: is the offline DTW explained in 3.2.3 based on Ellis approach. It computes the whole distorsion/similarity matrix and uses backtracing to find the aligned path.

- Online: is our particular approach to an online version of DTW. It simply computes the minimum cumulated cost in the current time, as is explained in 3.2.4

- Online with restrictions: as it is shown in 3.2.5 and 3.2.6 is an algorithm based on the online version but with two type of tunings; a global restriction and a tuning of the cost matrix of the DTW according to the estimation velocity.

- Online with correlation: is similar to the previous one but the estimation velocity is calculated in some particular times as is explained in 3.2.7.

For the evaluation of the results we use the Offline algorithm as a reference to compare the following approaches. As we want to affect the behavior of the DTW in a way that makes it more regular, we expect to lose some alignment precision. Thus, the best result that we can achieve is the one this algorithm gives. On the other hand, we expect the Online algorithm to be the worst due to its own nature. As it takes the maximum global similarity for each time it should be confused by repeated notes placed in different times. For the other two approaches we expect them to perform in a more regular fashion during time, with an slight improvement of the precision for the Online with correlation.

In this section the objective results using the MIREX metrics are reviewed. We also discuss some subjective results according to tempo naturality perception of the synthesized output for the different approaches.

4.2 Objective results

4.2.1 Overall precision

The overall precision rate for the four algorithms is presented in 4.1.

	Overall precision
Offline	94.98 %
Online	2.02 %
Online with restrictions	82.62 %
Online with correlation	84.67 %

Table 4.1: Overall precision results

As it is shown the best overall precision is achieved with the offline algorithm due to its "knowledge of the future". As the minimum path is computed having the whole audio, the decision of which is the optimal path is taken with a higher certainity. On the other hand, the Online approach gives a very low precision as expected, due to its "freedom" to pick the global minimum cumulated cost in each time. As expected, the approach using predecided time places where calculate the estimation velocity performs slightly better than using some arbitrary places. Although the precision of this two algorithms is not as high as in the Offline approach, it is still quite good.

4.2.2 Standard offset

As commented previously, the standard offset shows the standard time deviation according to the annotated ground truth. The results for the four evaluated algorithms are shown in 4.2. As expected, the lowest value is achieved with the offline algorithm and the highest with the online. In the

	Standard offset [ms]
Offline	47.02
Online	54.59
Online with restrictions	47.90
Online with correlation	51.15

Table 4.2: Standard offsets results

case of our two proposed algorithms the one using the correlation shows a slighlty highest value than the online with restrictions. This effect occurs probably due to the fact that in the online with correlation approach the estimation velocity is calculated in fewer places. Thus, the DTW performance is forced in a particular direction during a longer period, leading this way to slight time deviations.

4.2.3 Accuracy results

In 4.3 the other global measures are shown, as well as the previous ones. Precision corresponds to Overall precision, Miss corresponds to one percentage of missed notes, Missalign is the average misalignment in miliseconds, As.Offset is the average offset in miliseconds and Std Offsets corresponds to standard offset in miliseconds. As is shown in the table, the two proposed algorithms resulting from this research (Online with restrictions and Online with correlation) are able to align all the notes in the music collection. Moreover, the average offset is much lower than in the basic Online approach and quite close to the best result, achieved by the Offline algorithm.

	Precision	Miss	Missalign	Av.Offset	Std Offset
Offline	94.98	0	0.05	33.02	47.02
Online	2.02	0.12	0.90	40.06	54.49
Online with restrictions	82.62	0	0.17	32.64	47.90
Online with correlation	84.67	0	0.15	25.09	51.15

Table 4.3: Accuracy results

4.2.4 Regularity

The objective of this thesis is to achieve a more regular performance of the DTW algorithm. Before a perceptual evaluation of the results a comparison between the Offline approach and our final Online with correlation algorithm is shown in 4.1 and 4.2.

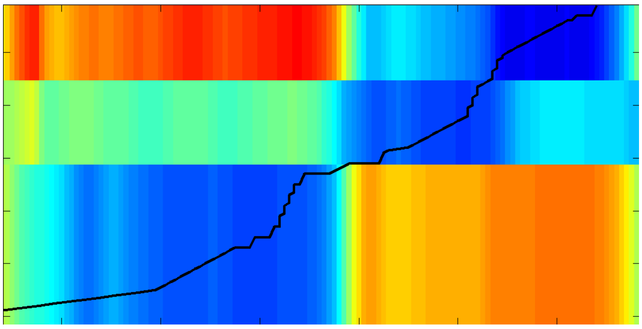


Figure 4.1: Estimated path using Offline algorithm

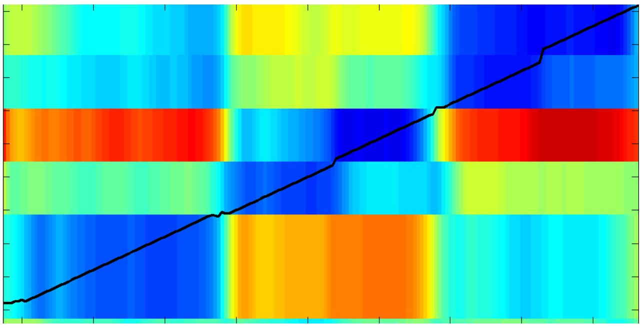


Figure 4.2: Estimated path using Online with correlation algorithm

The audio used to compare this two approaches is the first from the music collection commented in 3.2.2. In both figures the estimated path in a particular excerpt of time is shown on top of the confusion matrix. As we desired, the path in the Online with correlation approach is clearly more regular. This tendency is perceived along the whole collection. The Online with restrictions algorithm shows a similar behavior but is not as good in terms of precision and accuracy. Therefore, it seems that the algorithm performs as is expected for the goal of this research. Still, the regularity of the path is not directly linked to a more regular performance in terms. Thus, a perceptual test is peformed in the following section.

4.3 Subjective results

As commented in 3.3.4, we built a straightforward perceptual test to verify the tempo regularity of the proposed approach. For such experiment we synthesize all the audios from the used database commented in 3.3.2 using the tools related in 3.3.4 for the four algorithms related in 4.1. We also use the algorithm proposed in (Carabias et al., 2012) to compare the results to a state of the art approach to the problem. The audios were presented to five musicaly trained users (more than four years of musical studies). After listening to each audio, the users were asked to grade how "natural" it sounded in terms of tempo in a five points scale from one being the least regular to five being the most similar to a human performance. An average of the results for the whole database for each algorithm are shown in 4.4. As expected from previous objective results, the Online approach show the lowest grade and the Offline the best one. Even so, the interesting comparison is between the two proposed algorithms in this work and the one proposed by J. Carabias. In this case, we can see a slighlty improvement of the tempo regularity percieved by the users.

Algorithm	Average rate
Offline	3.82
Online	1
Online with restrictions	3.02
Online with correlation	3.40
J. Carabias 2012	2.85

Table 4.4: Subjective results

Chapter 5

CONCLUSIONS

The aim of this research is to adress the problem of audio to score alignment from a musical oriented point of view, meaning that we want an output that could be used as a musical accompaniment for the input sound with a musical sense of tempo. For such prupose we started with the implementation of a DTW-based algorithm based on the previous work from J.Carabias, that had the best results on the last MIREX Score Following task. Although this approach shows high precision rate results, once we synthesize the output the audio does not have a natural feeling in terms of tempo. This effect occurs because the evaluation criteria considers a correct onset as falling within a 200 ms window centered in the corresponding annotated onset. Thus, two consecutive correct onsets could be respectively at -100ms and +100ms, leading to non regular tempo performance of the output. The proposed method focus on the performance of the dynamic time warping, using a combination of restrictions to force the algorithm to perform in a more regular fashion. The restrictions are applied according to what we called estimation velocity, that is somehow related to the input sound tempo. To evaluate the perfomance of the approach several algorithms are presented considering different type of restrictions. The results show that the applied restrictions lead to a more regular performance of the alingment process while keeping still good results in terms of precision. Furthermore, subjective analysis indicate that the output of this approach seems to have more musical meaning and sounds more natural.

5.1 Contributions

According to the goals defined in the Introduction section, in the scope of this research the following contributions have been made;

- A review of state of the art approaches to score alignment, their techniques and feature extraction methods and their applications.
- Implementation of a basic online score alignment algorithm based on Dynamic Time Warping.
- A method to calculate the estimation velocity of the DTW performance.
- A method to restrict the performance of the DTW according to the estimation velocity.
- Implementation of a novel online score alignment algorithm based on previous knowledge about ”where” the estimation velocity calculation is done.
- Analysis of the perceptual differences between the synthesized output of different approaches to score alignment.

5.2 Future work

Based on the results of this study, the following aspects could be of interest for future studies in the field:

- Consider musicology knowledge for the ”anchor points” decision rewied in 3.2.7. As commented previously, musicians use tempo as an expressivity facet, changing it during the performance. Although this changes in tempo varies from one performance to another, there are some recurrent patterns that can be used to improve our algorithm.
- Development of an Automatic Musical Accompaniment for educational and musicological purposes.
- Research on ”naturality” in perception of tempo changes to improve the algorithm.

Bibliography

Arulampalam, M., Maskell, S., Gordon, N., and Clapp, T. (2002). A tutorial on particle filters for online nonlinear/non-Gaussian Bayesian tracking. IEEE Transactions on Signal Processing, 50(2):174–188.

Arzt, A. (2007). Score Following with Dynamic Time Warping - An Automatic Page-Turner. PhD thesis.

Cano, P., Loscos, A., and Bonada, J. (1999). Score-Performance Matching using HMMs. In Proceedings of the ICMC99.

Carabias, J. J., Rodriguez, F. J., Vera, P., Caba, P., Ca, F. J., and Ruiz, N. (2012). A real-time nmf-based score follower for MIREX 2012. (2).

Carabias-Orti, J., Rodriguez-Serrano, F., Vera-Candeas, P., Cañadas Quesada, F., and Ruiz-Reyes, N. (2013). Constrained non-negative sparse coding using learnt instrument templates for realtime music transcription. Engineering Applications of Artificial Intelligence, 26(7):1671–1680.

Cont, A. (2004). Improvement of Observation Modeling for Score Following. PhD thesis.

Cont, A. (2006). Real time audio to score alignment for polyphonic music instruments using sparse non-negative constraints and hierarchical hmms. In ICASSP 2006 Proceedings. 2006 IEEE International Conference, Toulouse.

Cont, A. (2007). Antescofo: anticipatory synchornization and control of interactive parameters in computer music. Technical report.

Dannenberg, R. B. (1985). An On-Line Algorithm for Real-Time Accompaniment. In Proceedings of the 1984 International Computer Music Conference, pages 193–198.

Dannenberg, R. B. (2007). An intelligent multitrack audio editor. In Proceedings of the 2007 International Computer Music Conference, volume II, pages II –89–94.

Dannenberg, R. B. and Raphael, C. (2006). Music score alignment and computer accompaniment. Comunication of the ACM, 49.

Davies, M. E. P., Degara, N., and Plumbley, M. D. (2009). Evaluation Methods for Musical Audio Beat Tracking Algorithms. (October).

de Cheveigne, A. and Kawahara, H. (2002). YIN, a fundamental frequency estimator for speech and music. The Journal of the Acoustical Society of America, 111(4):1917.

Dixon, S. (2005). Live tracking of musical perfomances using on-line time warping. In Conference on Digital Audio Effects, pages 1–6, Madrid.

Duan, Z. and Pardo, B. (2011). Soundprism : An Online System for Score-informed Source Separation of Music Audio. IEEE Journal Of Selected Topics In Signal Processing, 5(6):1205–1215.

Ewert, S., Pardo, B., Mueller, M., and Plumbley, M. D. (2014). Score-Informed Source Separation for Musical Audio Recordings: An overview. IEEE Signal Processing Magazine, 31(3):116–124.

Flossmann, S. and Widmer, G. (2011). Toward a multilevel model of expressive piano performance. In iInernational Symposium on Performance Science, number 2003.

Fritsch, J. (2013). Score informed audio source separation using constrained nonnegatuve matriz factorization and score synthesis. In ICASSP 2013, number May, pages 888–891, Vancouver.

Gemmeke, J. F., Virtanen, T., and Hurmalainen, A. (2011). Exemplar-Based Sparse Representations for Noise Robust Automatic Speech Recognition. IEEE Transactions on Audio, Speech, and Language Processing, 19(7):2067–2080.

Grubb, L. and Dannenberg, R. B. (1994). Automated Accompaniment of Musical Ensembles. In AAAI-94 Proceedings.

Heckerman, D. (1995). A Tutorial on Learning Bayesian Networks. Technical Report March, Communications of the ACM.

Hennequin, R., David, B., and Badeau, R. (2011). Score informed audio source separation using a parametric model of non-negative spectrogram. In ICASSP, 2011, number 1, pages 45–48, Prague. IEEE.

Hu, N., Dannenberg, R. B., and Tzanetakis, G. (2003). Polyphonic Audio Matching and Alignment for Music Retrieval. IEEE Workshops on Applications of Signal Processing to Audio and Acoustics, pages 185–188.

Joder, C., Essid, S., and Richard, G. (2013). Learning Optimal Features for Polyphonic Audio-to-Score Alignment. IEEE Transactions on Audio, Speech, and Language Processing, 21(10):2118–2128.

Lee, D. D., Laboratories, B., Hill, M., and Ý, H. S. S. (2000). Algorithms for Non-negative Matrix Factorization. In Proc. of Neural Information Processing Systems, number 1, Denver, USA.

Mcnab, R. J., Smith, L. A., Witten, I. H., Henderson, C. L., and Cunningham, S. J. (1996). Towards the Digital Music Library : Tune Retrieval from Acoustic Input. In Proceedings of the first ACM international conference on Digital libraries, number 1978, pages 11–18.

Murphy, K. P. (2001). An introduction to graphical models. (May):1–19.

Nantes, C. D. (2011). Algorithms for nonnegative matrix factorization with the β -divergence. Neural Computation, 23(3):2412 – 2456.

Orio, N. and Déchelle, F. (2001). Score following using Spectral Analysis and Hidden Markov Models. In ICMC 2001.

Otsuka, T., Nakadai, K., Takahashi, T., Ogata, T., and Okuno, H. (2011). Real-Time Audio-to-Score Alignment Using Particle Filter for Coplayer Music Robots. EURASIP Journal on Advances in Signal Processing, 2011(1):384651.

Prockup, M., Grunberg, D., Hrybyk, A., and Kim, Y. E. (2013). Orchestral Performance Companion: Using Real-Time Audio to Score Alignment. IEEE MultiMedia, 20(2):52–60.

Puckette, M. (1990). Explode, a user interface fopr sequencing and score following. In ICMC '90 Proceedings.

Puckette, M. and Jolla, L. (1995). Score following using the sung voice. In ICMC 1995, pages 1–8.

Puckette, M. and Lippe, C. (1992). Score following in practice. In Proceedings of the ICMC '92.

Rabiner, L. R. (1989). A tutorial on Hidden Markov Models and Selected Applications in Speech Recognition.

Raphael, C. (1999). Automatic segmentation of acoustic musical signals using hidden Markov models. IEEE Transactions on Pattern Analysis and Machine Intelligence, 21(4):360–370.

Raphael, C. (2001). A Bayesian Network for Real-Time Musical Accompaniment. Advances in Neural Information Processing Systems NIPS 14, page 14.

Raphael, C. (2006). A hybrid graphical model for aligning polyphonic audio with musical scores. Machine learning, 65(2-3):389–409.

Raphael, C. (2010). Music Plus One and Machine Learning. In Proceedings of the Twenty-Seventh iInternational Conference ICML 2010.

Suzuki, K., Ueda, Y., Ono, N., and Sagayama, S. (2006). Real-time audio to score alignment using locally-constrained dynamic time warping of chromagrams. MIREX, 2006:3–4.

Vercoe, B. (1984). The sinthetic performer. In ICMC '84 proceedings, pages 199–200.

Woodruff, J., Pardo, B., and Dannenberg, R. (2006). Remixing Stereo Music with Score-Informed Source Separation. In Proceedings of the 7th international Conference on Music Information Retrieval.