SYMBOLIC AND STRUCTRUAL REPRESENTATION OF MELODIC EXPRESSION

Christopher Raphael
School of Informatics and Computing
Indiana Univ., Bloomington

ABSTRACT

A method for expressive melody synthesis is presented seeking to capture the structural and prosodic (stress, direction, and grouping) elements of musical interpretation. The interpretation of melody is represented through a hierarchical structural decomposition and a note-level prosodic annotation. An audio performance of the melody is constructed using the time-evolving frequency and intensity functions. A method is presented that transforms the expressive annotation into the frequency and intensity functions, thus giving the audio performance. In this framework, the problem of expressive rendering is cast as estimation of structural decomposition and the prosodic annotation. Examples are presented on a dataset of around 50 folk-like melodies, realized both from hand-marked and estimated annotations.

1. INTRODUCTION

A traditional musical score represents music *symbolically* in terms of notes, formed from a discrete alphabet of possible pitches and durations. Human performance of music often deviates substantially from the score's literal interpretation, by inflecting, stretching and coloring the music in ways that bring it to life. *Expressive music synthesis* seeks algorithmic approaches to this expressive rendering task, so natural to humans.

There is really a great deal of past work on expressive synthesis — more than can be summarized here, though some of the leading authors give an overview of several important lines of work in [1]. Most past work, for example [2], [3], [4], as well as the many RENCON piano competition entries, for example [5] [6], has concentrated on piano music. The piano is attractive for one simple reason: a piano performance can be described by giving the onset time, damping time, and initial loudness of each note. Since a piano performance is easy to represent, it is easy to define the task of expressive piano synthesis as an estimation problem: one must simply estimate these three numbers for each note.

This work supported by NSF grants IIS-0739563 and IIS-0812244

In contrast, we treat here the synthesis of *melody*, which finds its richest form with "continuously controlled" instruments, such as the violin, saxophone or voice. This area has been treated by a handful of authors, including the KTH group [7], [8], as well as a number others, including a commercial singing voice system. Continuously controlled instruments simultaneously modulate many different parameters, leading to wide variety of tone color, articulation, dynamics, vibrato, and other musical elements, making it difficult to represent the performance of a melody. However, it is not necessary to replicate any of these familiar instruments to effectively address the heart of the melody synthesis problem. We will propose a minimal audio representation we call the theremin, due to its obvious connection with the early electronic instrument by the same name [9]. Our theremin controls only time-varying pitch and intensity, thus giving a relatively simple, yet capable, representation of a melody performance.

The efforts cited above include some of the most successful attempts to date. All of these approaches map observable elements in the musical score, such as note length and pitch, to aspects of the performance, such as tempo and dynamics. One example is the rule-based KTH system, which grows out of several decades of focused effort. In this system, each rule maps various musical contexts into performance decisions, which can be layered, so that many rules can be simultaneously applied. The rules were chosen, and iteratively refined, by a music expert seeking to articulate and generalize a wealth of experience into performance principles. In contrast, the work of Widmer [2], [4] takes a machine learning perspective by *automatically* learning rules from actual piano performances. We share the perspective of machine learning. In [4], phrase-level tempo and dynamic curve estimates are combined with the learned rule-based prescriptions, through a case-based reasoning paradigm. That is, this approach seeks musical phrases in a training set that are "close" to the phrase being synthesized, using the tempo and dynamic curves from the closest training example. As with the KTH work, the performance parameters are computed directly from the observable score attributes with no real attempt to describe any *interpretive* goals such as repose, passing tone, local climax, surprise, etc.

Our work differs significantly from these, and all other past work we know of, by explicitly trying to *represent the interpretation itself*. Previous work does not represent the interpretation, but rather treats the *consequences* of this in-

terpretation, such as dynamic and timing changes. We represent the interpretation in two ways. This first uses a tree-like structural decomposition that makes explicit various levels of repetition or parallelism in the melody. This idea is familiar from other work such as [3], though we introduce a framework for automatically estimating the structure. This approach has connections with [10], which finds phrase decompositions from symbolic music. Secondly, we introduce a hidden sequence of variables representing the prosodic interpretation (stress and grouping) itself, by annotating the role of each note in the larger prosodic context. We believe these representations are naturally positioned between the musical score and the observable aspects of the interpretation. Thus the separate problems of estimating the representations and generating the actual performance from the representations require shorter leaps, and are therefore easier, than directly bridging the chasm that separates score and performance.

2. THE THEREMIN

Our goal of expressive melody synthesis must, in the end, produce actual sound. We introduce here an audio representation we believe provides a good trade-off between expressive power and simplicity.

Consider the case of a sine wave in which both frequency, $f(t)$, and amplitude, $a(t)$, are modulated over time:

$$s(t) = a(t)\sin(2\pi \int_0^t f(\tau)d\tau). \qquad (1)$$

These two time-varying parameters are the ones controlled in the early electronic instrument known as the *theremin*. Continuous control of these parameters can produce a variety of musical effects such as expressive timing, vibrato, glissando, variety of attack and dynamics. Thus, the theremin is capable of producing a rich range of expression. One significant aspect of musical expression the theremin *cannot* capture is tone color — as a time varying sine wave, the timbre of the theremin is always the same. Partly because of this weakness, we have modified the above representation to allow tone color to change as a function of amplitude:

$$s(t) = \sum_{h=1}^{H} A_h(a(t), f(t))\sin(2\pi h \int_0^t f(\tau)d\tau) \qquad (2)$$

where the $\{A_h\}$ are hand-designed functions, monotonically increasing in the first argument. Thus our sound is still parametrized by $f(t)$ and $a(t)$, while we increase the perceived dynamic range.

3. REPRESENTING MUSICAL INTERPRETATION

There are, no doubt, more aspects of musical interpretation than can possibly be treated here. Palmer [11] gives a very nice overview of current thinking on this subject from the Psychology perspective. Broadly speaking, there are at least three important components to musical interpretation: conveying musical structure, and, in particular, the way it relates to the notion of *phrase*; musical prosody — the placing, avoidance, and foreshadowing of local (note-level) stress and the associated low-level groupings that follow; and musical affect such happy, sad, intense, agitated, etc. We will focus only on phrase structure and prosody here, acknowledging that this is only a piece of the larger interpretive picture.

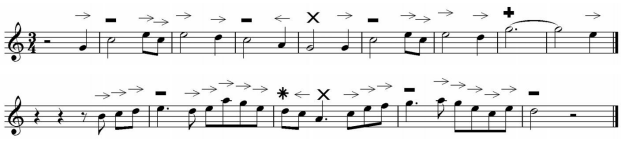

Figure 1. *Amazing Grace* (**top**) and *Danny Boy* (**bot**) showing the note-level labeling of the music using symbols from our alphabet.

The folk-like music we treat here is mostly composed of simple musical structure, with a high degree of repetition of rhythm, pitch contour, chord sequence, and other musical elements. Typically the hierarchical structure of these melodies is captured by simple tree structures, often involving binary groupings at various levels of grouping: it is no accident that 34 out of the 48 melodies in our dataset have 2^n measures for some n. Within this hierarchy, musical phrases correspond to "levels" of this tree. When a melody is not captured by a perfectly regular tree structure, it often corresponds to the concatenation of such regular trees. For instance, the familiar melody, *God Save the Queen*, may be described (2-2-2)+((2-2)-(2-2)) where each number represents a group of measures, '+' denotes concatenation and '-' denotes grouping. Thus the melody has 3 groups of two measures followed by a two levels of binary structure for the last eight measures. While there is a subjective component to the partition into *phrases*, the first 6 and last 8 measures seem like reasonable choices, perhaps splitting the last 8 measures into two 4-bar phrases. In this example phrase boundaries correspond exactly to measure boundaries, though often this is not the case. Thus we must also indicate the length of the "pickup" for each group of measures.

While conveying musical structure is an important part of expressive synthesis, the main focus of our effort here is on musical *prosody*. We introduce now a way of *representing* the desired musicality in a manner that makes clear interpretive choices and conveys these unambiguously. Our representation labels each melody note with a symbol from a small alphabet,

$$A = \{l^-, l^\times, l^+, l^\rightarrow, l^\leftarrow, l^*\}$$

describing the role the note plays in the larger context. These labels, to some extent, borrow from the familiar vocabulary of symbols musicians use to notate phrasing in printed music. The symbols $\{l^-, l^\times, l^+\}$ all denote stresses or points of "arrival." The variety of stress symbols allows for some distinction among the kinds of arrivals we

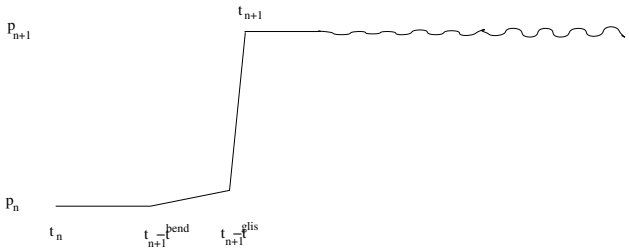

Figure 2. A graph of the frequency function, $f(t)$, between two notes. Pitches are bent in the direction of the next pitch and make small *glissandi* over the transitions.

can represent: l^- is the most direct and assertive stress; $l^\times$ is the "soft landing" stress in which we relax into repose; l^+ denotes a stress that continues *forward* in anticipation of future unfolding, as with some phrases that end in the dominant chord. Examples of the use of these stresses, as well as the other symbols are given in Figure 1. The symbols $\{l^\rightarrow, l^*\}$ are used to represent notes that move *forward* towards a future goal (stress). Thus these are usually shorter notes we pass through without significant event. Of these, $l^\rightarrow$ is the "garden-variety" passing tone, while l^* is reserved for the passing stress, as in a brief dissonance, or to highlight a recurring beat-level emphasis, still within the context of forward motion. Finally, the $l^\leftarrow$ symbol denotes receding movement as when a note is connected to the stress that precedes it. This commonly occurs when relaxing out of a strong-beat dissonance *en route* to harmonic stability. We will write $x = x_1, \ldots, x_N$ with $x_n \in A$ for the prosodic labeling of the notes.

These concepts are illustrated with the examples of *Amazing Grace* and *Danny Boy* in Figure 1. Of course, there may be several reasonable choices in a given musical scenario, however, we also believe that most labellings do *not* make interpretive sense and offer evidence of this is Section 7. Our entire musical collection is marked in this manner and available at

http://www.music.informatics.indiana.edu/papers/ismir09

4. FROM LABELING TO AUDIO

Ultimately, the prosodic labeling of a melody, using symbols from A, must be translated into the amplitude and frequency functions we use for sound synthesis. We have devised a deterministic mapping from our prosodically-labeled score to the actual audio parameter outlined here.

Our synthesis of $f(t)$ and $a(t)$ begins by modifying the literal interpretation of musical timing expressed in the score to include *ritardandi* (slowing down) at the ends of phrases. While we have not done so here, [3] recommends larger changes at higher levels of the phrase hierarchy, as expressed by our structural representation. We further modify $f(t)$ to include vibrato to long and stressed notes. Finally, we bend each pitch in towards the following pitch with a final *glissando* to encourage a sense of legato. Figure 2 shows a short piece of this pitch function over the two consecutive two notes.

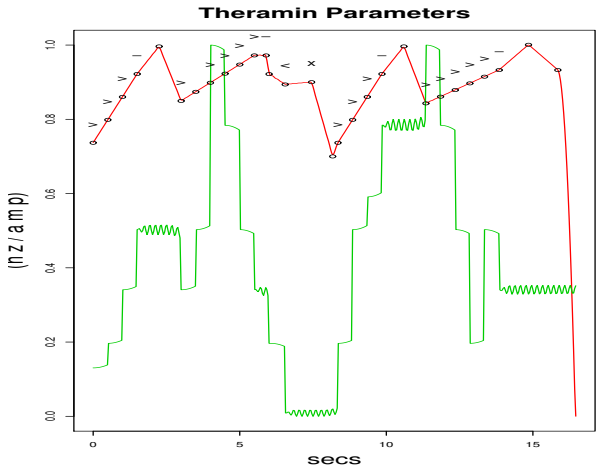

Figure 3. The functions $f(t)$ (green) and $a(t)$ (red) for the first phrase of *Danny Boy*. These functions have different units so their ranges have been scaled to 0-1 to facilitate comparison.

The heart of the transformation, however, is in the construction of the amplitude function $a(t)$. This function is created through a series of soft constraints that are placed on the amplitude defined at various "knot" locations over time. These constraints are taken from from the prosodically-annotated score and the structural representation. For instance, we want phrase beginnings, as indicated by the structural representation, to be low in amplitude; thus we add a quadratic penalty that encourages this characteristic. Similarly, we want stressed notes to be high in amplitude and add similar quadratic penalties to encourage this. In addition we want forward-moving notes to be increasing in amplitude, and thus add quadratic terms that encourage this relationship between a forward-moving note and its successor. Similar terms are added for receding notes. We then compute the values at the knot locations by minimizing the quadratic penalty function, and interpolate the resulting amplitudes at the knot locations. A more detailed presentation of this process is described in [12]. An example of both the $a(t)$ and $f(t)$ functions for a familiar examples are given in Figure 3.

5. HOW MUCH MUSICALITY DOES THE REPRESENTATION CAPTURE?

The theremin parameters, $f(t), a(t)$, and hence the audio signal, $s(t)$, depend entirely on the structural representation, the prosodic labeling, and the musical score, through the mapping described in Section 4. We want to understand the degree to which our representation captures musically important interpretive notions. To this end, we have constructed a dataset of about 50 simple melodies containing a combination of genuine folk songs, folk-like songs, Christmas carols, and examples from popular and art music of various eras. The melodies were chosen to be familiar, having simple chords, simple phrase structure, all at mod-

erate to slow tempo, and appropriate for *legato* phrasing. Examples include *Danny Boy*, *Away in a Manger*, *Loch Lomond*, *By the Waters of Babylon*, etc. These melodies were painstakingly hand-annotated with structure and prosody by the author.

We rendered these melodies into audio according to our hand-marked annotations and the process of Section 4. For each of these audio files we provide harmonic context by superimposing sustained chords, as indicated in the scores. The entire collection of symbolic melodies, along with rendered audio files, is available at the aforementioned web site.

We do observe some aspects of musical interpretation that are not captured by our representation. For example, the interpretation of *Danny Boy* clearly requires a climax at the highest note, as do a number of the musical examples. We currently do not represent such an event through our markup. It is possible that we could add a new category of stress corresponding to such a highpoint, though we suspect that the degree of emphasis is continuous, thus not well captured by a discrete alphabet of symbols.

Another occasional shortcoming is the failure to distinguish contrasting material, as in *O Come O Come Emanuel*. This melody has a Gregorian chant-like feel and should mostly be rendered with deliberate calmness. However, the short outburst corresponding to the word "Rejoice" takes on a more declarative affect. Our prosodically-oriented markup simply has no way to represent such a contrast of styles, though it is hinted at in the structural decomposition of ((3-3)-(3-3))+(2-2)+3.

There are, perhaps, some other general shortcomings of the interpretations, though we believe there is quite a bit that is "right" in them, especially considering the simplicity of our representation of interpretation. However, we hope readers will make independent judgments.

6. ESTIMATING THE INTERPRETATION

The essential goal of this work is to *algorithmically* generate expressive renderings of melody. Having formally represented our notion of musical interpretation, we can generate an expressive rendering by *estimating* this representation.

6.1 Estimating Phrase Structure

We estimate the structural decomposition of our melody by maximizing an objective function defined on the decomposition using dynamic programming. The approach begins by labeling each note subsequence containing two bar lines as a *terminal state*, and scoring the plausibility of each possible label for the subsequence (the score function will be discussed presently). We then proceed *inductively* to find the optimal labelings of progressively larger subsequences, ultimately terminating with a labeling for the entire melody.

Suppose we have have found the possible labelings of each note subsequence containing $m-1$ bar lines, and have computed the best-scoring derivation of each such labeled subsequence (the labels will be described below). We can find the optimal score of each label on each contiguous region containing m bar lines by piecing together various contiguous subsequences containing less than m bar lines. We allow three possible ways to do this, as follows

1. We can label a subsequence containing m bar lines as a *terminal* state, corresponding to a single grouping with no subdivisions. We label such a group of measures as m — the number of measures composing the group. The subsequence need not begin or end at a measure boundary.

2. If the number of measures, m, has a factor, f, in $\{2, 3, \ldots, 5\}$, we consider all partitions of the region into f contiguous regions each containing $k = n/f$ bar lines. For each such partition, we consider piecing together k identically labeled segments and labeling the result as $(k - k - \ldots - k)$. For instance, if we consider a region containing 8 bar lines and consider composing this region of two identically labeled contiguous regions, we could group regions labeled as either 4 or (2-2). Any such production would result in a region labeled as (4-4), denoting the binary split. We *cannot* combine two contiguous regions labeled as 4 and (2-2) to make a (4-4) region.

3. For the final production phase, which considers the complete collection of melody notes containing, say, M bar lines, we allow the previously-described productions as well as a *concatenation* operation. The concatenation pieces together any pair or triple of contiguous regions composing the complete melody. Such concatenations will be denoted as $A + B$ or $A + B + C$ where A, B, C are any possible labelings of the individual regions.

Each of these productions generates a score for the resulting labeling. When we use the terminal state label, we want the collection of measures to make sense as an isolated unit. Thus we will score such labels to reward relatively long final notes and chord changes at the following bar line.

When applying our factoring rule, we wish to group together note sequences that exhibit parallelism. The rhythmic parallelism between two note groups can be measured by the symmetric difference of the rhythms — the number of notes that do note "line up" when the bar lines are aligned. This measure rewards similar rhythmic structures and encourages groups to have the same pickup length. When more than two groups are considered, we can compute an average symmetric difference. We have used such average symmetric differences on rhythm, pitch, and chord to achieve an overall measure of parallelism. The score of a particular factor label will then be the sum of the individual labeled subsequence scores plus the score for overall parallelism.

The final production type is concatenation. Generally speaking, we wish to discourage such explanations, so we give a fixed penalty every time the concatenation operation

is invoked. Thus the score for a label involving concatenation is the sum of the individual scores, plus a parallelism score between the concatenated sections, plus the concatenation penalty.

With this description in mind, it is simple to find the overall best scoring labeling. After computing and scoring all possible labelings of regions containing m bar lines, we retain only the best scoring parse for each particular label — this is the essential idea of dynamic programming. Finally, when we consider the entire collection of notes, we choose the best scoring of all labelings as our structure estimate.

At present we have simply hand-chosen the score function and make no claims for the optimality of this choice. Both the automatic training and evaluation of this method are the focus of ongoing work. As an example, our algorithm recognized *O Come O Come Emmanuel* as ((3-3)-(3-3))+7 with each segment containing a quarter note pickup, showing an ability to recognize interesting asymmetries. Appropriately, most often we recognized simple binary structures to our melodies.

6.2 Estimating the Prosodic Labeling

Our estimation of the unobserved sequence of prosodic labels, $x_1, \ldots, x_N$, depends on various observables, $y_1, \ldots, y_N$, where the feature vector $y_n = y_n^1, \ldots, y_n^J$ measures attributes of the musical score at the nth note. The features we consider are surface-level attributes of the musical score. While a great many possibilities were considered, we ultimately culled the set to the metric strength of the onset position, as well as the first and second differences of note length, in seconds, and MIDI pitch.

Our fundamental modeling assumption views the label sequence, x, as a Markov chain, given the data, y:

$$\begin{aligned} p(x|y) &= p(x_1|y_1) \prod_{n=2}^{N} p(x_n|x_{n-1}, y_n, y_{n-1}) \quad (3) \\ &= p(x_1|y_1) \prod_{n=2}^{N} p(x_n|x_{n-1}, z_n) \end{aligned}$$

where $z_n = (y_n, y_{n-1})$. The intuition behind this assumption is the observation (or opinion) that much of phrasing results from a cyclic alternation between forward moving notes, $\{l^{\rightarrow}, l^{*}\}$, stressed notes, $\{l^{-}, l^{+}, l^{\times}\}$, and optional receding notes $\{l^{\leftarrow}\}$. Often structural boundaries occur when one moves from either stressed or receding states to forward moving states. Thus the notion of *state*, as in a Markov chain, seems to be relevant.

We estimate the conditional distributions $p(x_n|x_{n-1}, z_n)$ for each choice of $x_{n-1} \in A$, as well as $p(x_1|y_1)$, using our labeled data. We will use the notation

$$p_l(x|z) \stackrel{\text{def}}{=} p(x_n = x|x_{n-1} = l, z_n = z)$$

for $l \in A$. In training these distributions we split our score data into $|A|$ groups, $D_l = \{(x_{li}, z_{li})\}$, where D_l is the collection of all (class label, feature vector) pairs over all notes that immediately follow a note of class l.

	l^{*}	$l^{\rightarrow}$	$l^{\leftarrow}$	l^{-}	$l^{\times}$	l^{+}	total
l^{*}	135	112	0	18	2	0	267
$l^{\rightarrow}$	62	1683	8	17	0	0	1770
$l^{\leftarrow}$	3	210	45	6	2	0	266
l^{-}	49	48	4	103	15	0	219
$l^{\times}$	5	32	2	65	30	0	134
l^{+}	0	3	0	12	3	0	18
total	254	2088	59	221	52	0	2674

Figure 4. Confusion matrix of errors over the various classes. The rows represent the true labels while the columns represent the predicted labels. The block structure indicated in the table shows the confusion on the coarser categories of stress, forward movement, and receding movement

We model the $p_l(x|z)$ distributions using the classification tree methodology of CART [13]. That is, for each D_l we begin with a "split," $z^j > c$ separating D_l into two sets: $D_l^0 = \{(x_{li}, z_{li}) : z_{li}^j > c\}$ and $D_l^1 = \{(x_{li}, z_{li}) : z_{li}^j \leq c\}$. We choose the feature, j, and cutoff, c, to achieve maximal "purity" in the sets D_l^0 and D_l^1 as measured by the average entropy over the class labels. We continue to split the sets D_l^0 and D_l^1, splitting their "offspring," etc., in a greedy manner, until the number of examples at a tree node is less than some minimum value. Our estimate $\hat{p}_l(x|z)$ is then computed by finding the terminal tree node associated with z and using the empirical label distribution over the class labels $\{x_{li}\}$ whose associated $\{z_{li}\}$ fall to the same terminal tree node.

Given a piece of music with feature vector $z_1, \ldots, z_N$,we can compute the optimizing labeling

$$\hat{x}_1 \ldots, \hat{x}_N = \arg \max_{x_1, \ldots, x_N} \hat{p}(x_1|y_1) \prod_{n=2}^{N} \hat{p}(x_n|x_{n-1}, z_n)$$

using dynamic programming.

7. RESULTS

We estimated a labeling for each of the $C = 48$ pieces in our corpus by training our model on the remaining $C - 1$ pieces and finding the most likely labeling, $\hat{x}_1, \ldots, \hat{x}_N$, as described above. When computing the most likely labeling for each melody in our corpus we found a total of 678/2674 errors (25.3%) with detailed results as presented in Figure 4.

The notion of "error" is somewhat ambiguous, however, since there really is no correct labeling. In particular, the choices among the forward-moving labels: $\{l^{*}, l^{\rightarrow}\}$, and stress labels: $\{l^{-}, l^{\times}, l^{+}\}$ are especially subject to interpretation. If we compute an error rate using these categories, as indicated in the table, the error rate is reduced to 15.3%.

One should note a mismatch between our evaluation metric of recognition errors with our estimation strategy. Using a forward-backward-like algorithm it is possible to

compute $p(x_n|y_1,\ldots,y_N)$. Thus if we choose

$$\bar{x}_n = \arg\max_{x_n \in A} p(x_n|y_1,\ldots,y_N),$$

then the sequence $\bar{x}_1,\ldots,\bar{x}_N$ minimizes the expected number of estimation errors

$$E(\text{errors}|y_1,\ldots,y_N) = \sum_n p(x_n \neq \bar{x}_n|y_1,\ldots,y_N)$$

We have not chosen this latter metric because we want a *sequence* that behaves reasonably. It the sequential nature of the labeling that captures the prosodic interpretation, so the most likely sequence $\hat{x}_1,\ldots,\hat{x}_n$ seems like a more reasonable choice.

In an effort to measure what we believe to be *most* important — the perceived musicality of the performances — we performed a small user study. We took a subset of the most well-known melodies of the dataset and created audio files from the random, hand, and estimated annotations. The estimated annotations were produced using ground truth for the structure while estimating the prosodic labelings. We presented all three versions of each melody to a collection of 23 subjects who were students in our University's music school, as well as some other comparably educated listeners. The subjects were presented with random orderings of the three versions, with different orderings for each user, and asked to respond to the statement: "The performance sounds musical and expressive" with the Likert-style ratings 1=strongly disagree, 2=disagree, 3=neutral, 4=agree, 5=strongly agree, as well as to rank the three performances in terms of musicality (the ranking does not always follow from the Likert ratings). Out of a total of 244 triples that were evaluated in this way, the randomly-generated annotation received a mean score of 2.96 while the hand and estimated annotations received mean scores of 3.48 and 3.46. The rankings showed no preference for the hand annotations over the estimated annotations ($p = .64$), while both the hand and estimated annotations were clearly preferred to the random annotations ($p = .0002$, $p = .0003$).

Perhaps the most surprising aspect of these results is the high score of the random labelings — in spite of the meaningless nature of these labelings, the listeners were, in aggregate, "neutral" in judging the musicality of the examples. We believe the reason for this is that musical prosody, accounts for only a portion of what listeners respond to. All of our examples were rendered with human-supplied structural representations and the same sound engine of Section 4 which tries to create a sense of smoothness in the delivery with appropriate use of vibrato and timbral variation. We imagine that the listeners were partly swayed by these aspects, even when the use of prosody was not satisfactory. The results also show that our estimation produced annotations that were, essentially, as good as the hand-labeled annotations. This demonstrates a success of our research. The computer-generated interpretations clearly demonstrate some musicality with an average listener rating of 3.46 — halfway between "neutral" and "agree." However, there is considerable room for improvement.

The melodies were also rendered using structural representations estimated as in Section 6.2, thus leaving the entire musical interpretation to the computer. The audio files documenting this experiment are available on the aforementioned web site.

8. REFERENCES

[1] Goebl W., Dixon S., De Poli G., Friberg A., and Bresin R. and Gerhard Widmer. *Sense in expressive music performance: Data acquisition, computational studies, and models*, chapter 5, pages 195–242. Logos Verlag, Berlin, may 2008.

[2] Widmer G. and Goebl W. Computational models for expressive music performance: The state of the art. *Journal of New Music Research*, 33(3):203–216, 2004.

[3] Todd N. P. M. The kinematics of musical expression. *Journal of the Acoustical Society of America*, 97(3):1940–1949, 1995.

[4] Widmer G. and Tobudic A. Playing Mozart by analogy: Learning multi-level timing and dynamics strategies. *Journal of New Music Research*, 33(3):203–216, 2003.

[5] Hiraga R., Bresin R., Hirata K., and Katayose H. Rencon 2004: Turing Test for musical expression, Proceedings of the 2004 Conference on New Interfaces for Musical Expression (NIME04), 120–123, 2004.

[6] Hashida Y., Nakra T., Katayose H., and Murao Y. Rencon: Performance Rendering Contest for Automated Music Systems, Proceedings of the 10th Int. Conf. on Music Perception and Cognition (ICMPC 10), Sapporo, Japan, 53-57, 2008.

[7] Sundberg J. The KTH synthesis of singing. *Advances in Cognitive Psychology. Special issue on Music Performance*, 2(2-3):131–143, 2006.

[8] Friberg A., Bresin R. and Sundberg J. Overview of the KTH rule system for musical performance. *Advances in Cognitive Psychology*, 2(2-3):145–161, 2006.

[9] Roads C. *The Computer Music Tutorial*. MIT Press, 1996.

[10] Bod R. A Unified Model of Structural Organization in Language and Music. *Journal of Artificial Intelligence Research*, 17:289-308, 2002.

[11] Palmer C. Music Performance. *Annual Review Psychology*, 48:115-138, 1997.

[12] omitted for review Representation and Synthesis of Melodic Expression. *Proc. of Int. Joint Conf. on Art. Int. (IJCAI))*, to appear.

[13] Breiman L., Friedman J., Olshen R. and Stone C. *Classification and Regression Trees*. Wadsworth and Brooks, Monterey, CA, 1984.