A DISCRETE FILTER BANK APPROACH TO AUDIO TO SCORE MATCHING FOR POLYPHONIC MUSIC

Nicola Montecchio, Nicola Orio
Department of Information Engineering
University of Padova
{nicola.montecchio,nicola.orio}@dei.unipd.it

ABSTRACT

This paper presents a system for tracking the position of a polyphonic music performance in a symbolic score, possibly in real time. The system, based on Hidden Markov Models, is briefly presented, focusing on specific aspects such as observation modeling based on discrete filterbanks, in contrast with traditional FFT-based approaches, and describing the approaches to decoding. Experimental results are provided to assess the validity of the presented model. Proof-of-concept applications are shown, which effectively employ the described approach beyond the traditional automatic accompaniment system.

1. INTRODUCTION

The concept of *audio to score alignment* refers to the ability of a system to align a digital audio signal recorded from a music performance with its score. More precisely, given a recording of a music performance and its score, the aim of such alignment system is to match each sample of the audio stream with the musical event it belongs to. There are a number of possible applications of such technology, ranging from the "automatic accompanist", a software allowing solo players to practice their part while the computer plays the orchestral accompaniment, to tools for musicological analysis or augmented audio access.

Most systems currently used for audio to score alignment are based on statistical models. In particular Hidden Markov Models (HMMs) [1, 5], possibly with hybrid approaches that make use of Bayesian networks and HMMs [8] or Hidden Hybrid Markov / semi-Markov chains [3].

In this paper we propose an HMM-based system that focuses on handling highly polyphonic music through the use of a filterbank approach.

2. MODEL DESCRIPTION

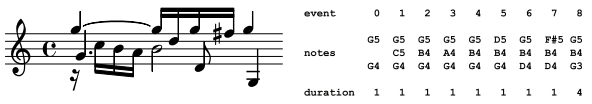
The main idea of the proposed approach is that the most relevant acoustic features of a music performance can be modeled statistically as observations of a Hidden Markov Model (HMM). The process of performing a music work can be regarded as stochastic because of the freedom of interpretation, yet the knowledge of the work that can be obtained from the score can be exploited to model the possible performances. In the presented system, a HMM is built according to the data contained in the music score. The incoming audio signal is divided into frames of fixed length, with every frame corresponding to one time step of the HMM; the HMM performs a transition every time a new audio frame is observed and the advancement of the performance in the score is tracked by performing the decoding of the HMM. The crucial point is the definition of the graph topology and the observation modeling while decoding is performed with well-known algorithms.

2.1 Score Graph Modeling

The score modeling step aims at obtaining a graph structure representing the music content of the score. In particular, a score is represented as a sequence of events, implying that it can be transformed into a simple graph where states are connected as in a chain. Two levels of abstraction can be distinguished in the resulting graph: a *score level* modeling the macro-structure of the piece, that is the sequence of music events, and an *event level* dealing with the structure of each music event; the distinction between the two reflects the conceptual separation between different sources of mismatch: the former deals with possible errors both by the musicians and in the score, while the latter models the duration and the acoustic features of each event, which vary depending on interpretation, instrumentation, recording conditions and other factors.

2.1.1 Score Parsing

The first step in building the HMM graph is the transformation of the symbolic score into a sequence of events. In the case of a monophonic score, all the notes and explicit rests correspond to an event, while events in a polyphonic score are bounded by any single onset and offset of all the notes that are played by the various instruments/voices (see Figure 1). Due to the large availability of already transcribed music, MIDI has been used as the score representation format although, being provided by end users, most of the MIDI files contain transcription errors that may influence the alignment effectiveness.



(a) Original score (b) Event sequence

Figure 1. Score representation

2.1.2 Graph Topology – Score Level

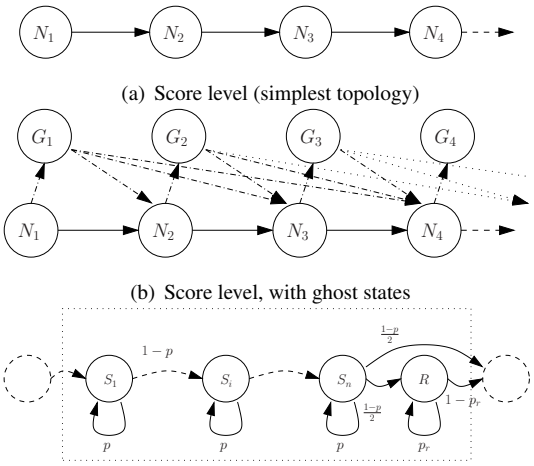
In its simplest form, the topology of the score level graph directly represents the succession of events: the states, each corresponding to a single music event, form a linear chain, as seen in Figure 2(a). This approach has no explicit model for local differences between the score representation and the actual performance that has to be aligned, thus the overall alignment can be affected by local mismatches. For instance, a skipped event, which should create only a local misalignment, can extend its effect also when subsequent correct events are played resulting in larger differences in the alignment.

In order to overcome these problems, a special type of states was introduced, namely *ghost states* – as opposed to *event states*, which correspond to real events in the music work. Ghost states were proposed in [4]. The basic graph topology is modified so that each event state can perform a transition to an associated ghost state, which in turn can perform either a self-transition or a forward transition to subsequent event states. The final representation is made of two parallel chains of nodes, as shown in Figure 2(b). This approach can model local differences between the score and the performance, because in this case the most probable path can pass through one or more ghost states during the mismatch and realign on the lower chain when the performance matches again the score. The transition probabilities from event states to corresponding ghost states are typically fixed, while the transition probabilities from a ghost state to subsequent event states follow a decreasing function of distance: this resembles the idea of *locality* of a mismatch due to an error.

2.1.3 Graph Topology – Event Level

The event level models the expected acoustic features of the incoming audio signal. Every state of this level is modeled as a chain of n *sustain states*, each having a self-loop probability p, possibly followed by a *rest state*, as shown in Figure 2(c). Sustain states model the features of the sustained part of an event, while rest states model the possible presence of silence at the end of each event that can be due to effects such as staccato playing style. As described in [7], the probability of having a segment duration d is modeled by a negative binomial distribution, with expected value $\mu = \frac{n}{1-p}$ and variance $\sigma^2 = \frac{np}{(1-p)^2}$. The duration of an event is modeled by setting the values of n and p accordingly; in particular μ is set equal to the event duration in the score.

Two cases can be distinguished depending on the choice of having n fixed or variable. In the former case event duration is modeled by self-loop probability. This approach is easy to implement and with a small n the total number of states in the graph is relatively small and proportional to the number of events; on the other hand the variance of the distribution changes with events duration. The latter case allows for a more precise modeling of event duration. It is reasonable to compute n and p in order to have $\sigma^2 = k\mu$, where p is constant for all the events, and the only parameter responsible for the event duration is the number of sustain states, of which the total number is thus proportional to the duration of the score.



(a) Score level (simplest topology)

(b) Score level, with ghost states

(c) Event level

Figure 2. Graph topologies

2.2 Modeling the Observations

The fundamental assumption of the model is that states of the event level emit the expected acoustic features of the incoming signal. Because polyphonic pitch detection is still unreliable, the signal itself is not analyzed, instead its harmonic features are compared to the expected features of the emissions of the HMMs.

2.2.1 Sustain States

The core feature used by the observation modeling of sustain states is the similarity, for each audio frame, between the spectrum of the incoming signal and an ideal spectrum of the sustain state that is being considered. Sophisticated techniques have been proposed making use of specific knowledge of instrument timbre [2]. Although very effective in specific situations, such as contemporary music performances where the instruments can be sampled, this kind of approach is not suitable for the general case where the instrument cannot be known in advance from the score.

Typically, spectrum analysis is done via the Fast Fourier Transform: the energies for the various frequency bands are computed by summing the energies in the appropriate FFT bins. The problem with this approach is that the linear frequency resolution of the FFT leads to a significant loss of precision in the lower frequency range. While the situation is partially compensated by upper harmonics a differ-

ent strategy can nevertheless improve the performances of a system.

In our approach, the frequency resolution problem is handled using a bank of discrete filters. In particular, each one is a second order filter of the form

$$H_i(z) = \frac{(1-r_i)\sqrt{1-2r_i\cos(2\theta_i)+r_i^2}}{(1-r_i e^{-j\theta_i}z^{-1})(1-r_i e^{j\theta_i}z^{-1})} \quad (1)$$

which has unit gain at θ_i (the normalized nominal frequency of the i-th note), and allows, by changing the pole radius r_i, to set the filter bandwidth; each filter output is then routed to a delay line in order to compensate for the different group delays: assuming that each filter has the same bandwidth in semitones, the filters corresponding to the lowest notes have a much higher group delay than the highest ones. We assume that this delay, which can be removed off-line or compensated in real time applications, is to be preferred to a lack of frequency resolution for lower notes. A comparison of FFT and Filterbank analysis is presented in Section 3.3.

The observation probability of a note is computed by partitioning the spectrum into frequency bands, with each band corresponding to a note in the music scale. Let E_i^f be the energy of the i-th filter output signal in the current frame, i.e. $E_i^f = \sum_t y_i^2(t)$; the energy E_i^n corresponding to the i-th note can be defined as

$$E_i^n = \sum_j E_{i+h(j)}^f \quad (2)$$

where $w_j = 1$ and $h(j)$ is a simple map between the index of a harmonic and the corresponding note index. In this very simple instrument model, the energy for the note C3 is computed as the sum of the energies for the filters corresponding to the notes C3, C4, G4, C5, E5, and so on.

The observation probability for the i-th sustain state is computed as

$$b_i^{(s)} = F(\frac{E_i^n}{E_{\text{tot}}}) \quad (3)$$

where E_i is the energy in the expected frequency bands and E_{tot} is the total energy of the audio frame. $F(\cdot)$ is the unilateral exponential probability density function

$$F(x) = \frac{e^\lambda}{e^\lambda - 1}\lambda e^{\lambda(x-1)} \quad 0 \le x \le 1 \quad (4)$$

Other similarity functions can be applied with similar results, in particular the cosine distance between the vector representations of the simple instrument model used to compute E_i and the filter output energies.

While the above approach is robust enough for monophonic alignment, the complexity of polyphony makes it preferable to apply a different weighting of the harmonics in the instrument model. A possible solution is to modify Equation 2 by adding decreasing weights to the note harmonics to reflect a more realistic instrument model. When filters overlap for some harmonics of different notes, the weight assigned to that harmonic in the instrument model can be either the sum or the maximum of the individual weights; the latter solution seems to perform better, and the intuitive explanation is that typical scores do not contain precise information about the loudness of each note/part, so a simpler model is more general.

2.2.2 Rest States

The observation probability for the i-th rest state is computed as a decreasing function of the ratio of the current audio frame energy over a reference threshold representing the maximum signal energy.

$$b_i^{(r)} = F(\frac{E_{\text{tot}}}{E_{\text{thres}}}) \quad (5)$$

The threshold is adaptive, to compensate for possible differences in the overall recording volume of different input streams.

2.2.3 Ghost States

A simple approach for modeling the observations of ghost states is to assign a fixed value to the observation probabilities, because these states are meant to provide a sort of "emergency exit" for local matches. The approach can be improved by computing the observation probability for the i-th ghost state as:

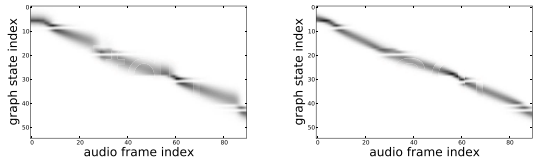
$$b_i^{(g)} = \sum_{j=i}^{i+k} w_i(j) b_j^{(s)} \quad (6)$$

that is, a weighted sum of the sustain observation probabilities of the following event states, where $w_i(\cdot)$ is a decreasing discrete distribution function and its presence is motivated by the fact that, intuitively, in case of wrong or skipped notes, the notes actually played would probably be close to the expected ones. In case of errors in the score, the weighting function induces the system to quickly realign on near notes.

2.3 Decoding Strategies

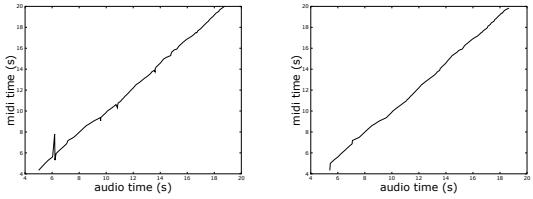
The proposed system exploits the decoding algorithms described in [6], depending on the application context, namely *forward decoding* and *forward-backward decoding*. These strategies determine, at each time interval, the *most probable state*, without forcing the decoded sequence of states to actually be the *most probable sequence* of states as is the case for *Viterbi decoding*. Preliminary tests showed that the system recovers more quickly, because the decoded sequence does not need to be a feasible state sequence.

Figure 3 compares a typical evolution of the state probabilities for the forward and forward-backward decoding algorithms. The latter is characterized by a more precise evolution, a highly desirable behavior in the case of subsequent events with the same set of harmonics: if no modeling of a note attack is employed – as is the case with the current version of the system – and the rest states at the end of the lower level event chain do not help discriminating the events, the evolution of forward-backward decoding automatically assigns to the events a duration in the alignment which is proportional to the duration in the score.



(a) Forward decoding (b) Forward-backward decoding

Figure 3. Evolution of state probabilities



(a) Forward decoding (b) Forward-backward decoding

Figure 4. Typical alignment evolution

3. EXPERIMENTAL RESULTS

The evaluation of an audio to score alignment system is a difficult task, mainly because of the lack of a manually aligned test collection of polyphonic music. For instance, the MIREX test collection is not publicly available because of copyright reasons and it contains mainly monophonic recordings. For this reason, two test collections have been prepared, the former made up of single-instrument polyphonic pieces and chamber music and the latter comprising excerpts of more complex orchestral works. A experimental comparison of the FFT and Filterbank analysis approaches is presented using recordings of tuba and cello music, characterized by a low frequency content.

3.1 Single Instrument and Chamber Music Collection

The audio collection is made up of excerpts from well known piano, violin, and chamber music works [1] extracted from CD and home recordings; the MIDI files were downloaded from the Internet. The files in the collection have been chosen so that the complexity of their polyphony is representative of pieces which could be realistically used in a typical automatic accompaniment system, with real time requirements. The resulting alignments were manually checked, visually inspecting the mismatches and aurally verifying them by listening to a stereo recording containing the original piece and a synthesized version generated from the alignment data on different channels.

Out of 20 test recordings, none caused the system to get lost, but in one case the alignment was very unstable (it was always in proximity of the "true" alignment but never precise) so its contribution will not be considered. For the other recordings the mismatches were classified according to their duration as either brief (shorter than two seconds) or long (larger time intervals, although never more than 10 seconds); the former type of mismatches occurred 41 times while the latter 10 times, mostly on complex passages of polyphonic material. Example alignments can be viewed and heard in the authors' home pages [2], where more detailed statistics can also be found.

Because of the real time requirements, the forward decoding algorithm was used to compute the alignments. If real time is not a constraint, usually forward-backward decoding gives better results, in which many of the glitches in the forward-decoded alignment are eliminated. Such an example is shown in Figure 4.

All the alignments have been performed using the same model parameters; further experiments showed that some improvements can be obtained by assigning different weights to the harmonics in Equation 2 for piano and string works. Essentially, the different weighting reflects the suitability of a more refined instrument model, in particular the piano model is characterized by more rapidly decaying overtones than the string model.

3.2 Orchestral Music Collection

The orchestral music collection comprises 48 excerpts of 40 seconds from CD recordings of symphonic works [3] ; the MIDI scores are generally much less accurate than the ones used in the chamber music collection.

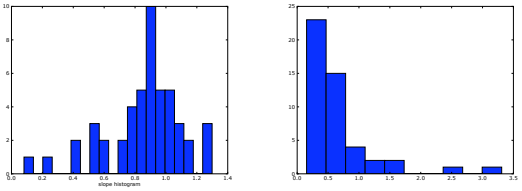
A simple evaluation methodology was devised in order to present results for this collection. The output of the alignment system for a single performance/score couple is a list of value pairs in the form [*audiotime*,*miditime*]. Once all the performances in a collection are aligned to their corresponding score, these alignments are analyzed to extract a measure of precision based on the average deviation of the alignment data from the best fitting line. This measure is based on the hypothesis that an orchestra plays more or less *a tempo*, at least in short time intervals, thus a graphic representation of the alignment should follow a straight line. While this is clearly a potentially incorrect assumption, the suitability of the particular performances in the test collection was verified by the authors. The best fitting line computed from the alignment data is thus assumed to be the correct alignment; Δ_{avg} is defined as the average deviation of the alignment data point from the best fitting line. Under the assumption of a performance characterized by a steady tempo, the lower is Δ_{avg} the higher is the alignment accuracy. This evaluation methodology was not used for the chamber music collection because the tempo was not steady enough.

Figure 5 shows the histograms of the slope and Δ_{avg} distributions for the best fitting lines obtained from the alignments. The tempo of the recorded performances and of the respective MIDI files are roughly comparable, so

[1] Bach: Italian Concerto, Goldberg Variations, Chaconne from the Violin Partita in D minor; Beethoven: Piano Sonata op. 13, String Quartet op. 18 n. 1; Mozart: Piano Sonata KV333; Ravel: String Quartet; Schubert: Quartettsatz D703; Schumann: Waldszenen op. 82.

[2] `http://www.dei.unipd.it/~montecc2/ismir09/`

[3] Beethoven: Symphonies n. 3, 7, 9; Haydn: Symphony n. 104; Mendelssohn: Symphony n. 4; Mozart: Symphonies and Serenades K136, K412, K525, K550; Vivaldi: The Four Seasons.



(a) Slopes histogram (b) Δ_{avg} histogram

Figure 5. Orchestral collection alignment results

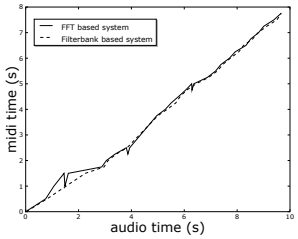


Figure 6. Comparison of FFT and Filterbank approaches

the expected histogram of the slopes should be centered around 1; an alignment can thus be safely considered incorrect when slope values are outside the interval $(0.6, 2)$. This simple assumption allows to quickly interpret the graphical results and deduce that the performance of the system with orchestral music is, as expected, clearly worse than the case for single instrument or chamber music, in which all alignments were essentially correct. Manual inspection of the results showed that the correct alignments were 36; for those, the average Δ_{avg} was 0.47.

A closer analysis pointed out that in the correct and incorrect sets of alignment the elements are homogeneous with respect to the music work, e.g. all Vivaldi's and most of Mozart's music was correctly aligned while most of Beethoven's were not. The reason for this was found out to be the fact that in the recordings of Beethoven's works the reference pitch was slightly higher than the standard 440 Hz for A4; correcting this setting considerably improved the results for Beethoven's music. This situation is a clear example of how a single set of parameters is not suitable for all the possible situations, but this is typically not a requirement: in the offline case multiple alignments can be performed and only the best one, according to the simple heuristics discussed above, can then be presented to the user, while when real time is required, it is reasonable to assume that the system parameters can be adjusted using previous rehearsals as reference.

In the above results, the forward decoding algorithm was used to compute the alignments; the reason is that the forward-backward algorithm turned out to be less robust for aligning performances where an alignment computed with forward decoding was not precise.

3.3 Comparison of FFT and Filterbank analysis

Several experiments were performed on a small collection of recordings of tuba and cello music, to show the advantages of discrete Filterbank analysis over traditional FFT for observation modeling on music characterized by a low frequency content. The recordings were aligned manually in order to count the number of wrongly recognized or skipped notes. Of 105 total events, the FFT based system did not recognize 12 and skipped 1, while the Filterbank based system did not recognize only 4 notes and skipped none. It should be noted that in almost all cases of not recognized notes both system realigned on the correct note immediately, and that the parameters of the systems were not tuned for this particular situation, so that better performances can be expected; forward decoding was used to simulate a real-time operation. The alignments of the worst performing recording are shown in Figure 6.

4. APPLICATIONS

Two applications are presented that make use of audio to score alignment technology for music analysis tasks.

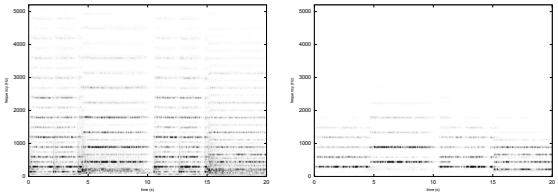
4.1 AudioZoom

AudioZoom is a software for the auditory highlight of single instruments in a complex polyphony. The basic idea is that the alignment can help dividing a polyphonic music performance into its individual components: the general problem is known as *source separation*, which is usually defined *blind* when it is assumed that almost no information is available about the role of each source. In our case, having the score as a reference, the system has a complete knowledge about the notes played, at each instant, by all the instruments. The user, typically a teacher who may exploit this tool to highlight particular instruments or passages to students that are not able to follow a complex score, can select one or more instruments, one or more particular musical themes or patterns, or any combination, and the system can selectively amplify the chosen elements.

The final effect is to put on the front, or zooming, the interested elements. A prototype of *AudioZoom* has been developed, based on a bank of bandpass filters centered around the harmonics of a selected instrument, using an approach similar to the instrument model described in Section 2.2.1. The user selects one channel from the MIDI file that represents the score, and the system aligns the different filterbanks with the audio recording. An example of the effect of AudioZoom, applied to the viola part of the beginning of Haydn's Symphony n. 104, is shown in the sonograms of Figure 7.

4.2 Interpretation Analysis

Analyzing different interpretations of a music work is a central activity of musicological analysis. Of all the features that characterize a personal rendition, tempo is probably the most perceivable one. The alignment of two audio performances allows to compare the relative tempos, but neither can be considered as a reference since no interpretation can be neutral. It can be noted that the concept of neutral interpretation is itself not well defined.



(a) Original excerpt (b) Zoomed excerpt

Figure 7. Effect of *AudioZoom*

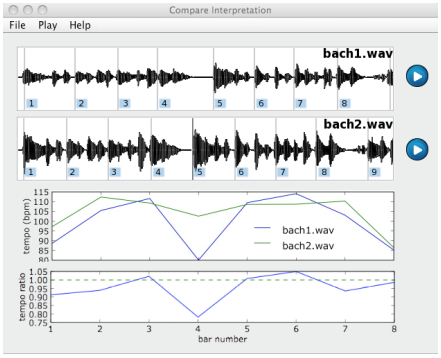


Figure 8. Different interpretations of the same piece

The alignment of two interpretations to the score allows a musicologist to draw some considerations on the different interpretations, for instance by comparing the instantaneous tempo at each bar. Figure 8 shows an early prototype of a tool for the comparison of different performances, in which two interpretations of the beginning of J. S. Bach's Italian Concerto are juxtaposed using the measures in the score as a reference. Clearly, the prototype can be extended by representing the differences in loudness, the use of *accelerandi* and *rallentandi* or more complex features related to timbre perception.

5. CONCLUSIONS AND FUTURE WORK

A system is proposed for the alignment of an audio performance with a score. The system is based on the use of filterbanks to extract pitch related information from the performances. Comparative evaluations with previous versions of the system showed that observation modeling based on discrete filterbanks has some advantages with respect to the simpler FFT approach, resulting in higher effectiveness. In general, evaluation showed that the approach can be effectively applied to real application scenarios; many areas however can be improved, and below we propose some research directions which seem the most promising.

A clear priority is the creation of a collection which comprises precise manual alignments, in order to properly evaluate the effectiveness of the approach but also to train the model parameters in a rigorous way. This is a very time-consuming task, requiring music experts and specific annotation tools for properly marking the matches between the events in the scores and the corresponding time instants in the recordings. The only viable solution in our opinion is to involve other research teams in building a shared collection of reasonable size; such collaborative effort would also help in devising appropriate data and evaluation methodologies for alignment system. A good starting point is the collection used for the MIREX campaigns, which should be improved adding polyphonic scores and a clearer time reference for the alignment evaluation.

The introduction of a refined modeling for the attack of notes is desirable for many instruments with percussive attacks – in particular the piano – to better handle repeated notes, but with the appropriate decoding strategies this issue is not critical. Another improvement regards the modeling of complex events, such as trills or *glissandi*, which are hard to extract from MIDI files, resulting in potentially less effective models.

6. ACKNOWLEDGMENTS

The work has been partially supported by a grant of the University of Padova for the project "Analysis, design, and development of novel methodologies for the study and the dissemination of music works".

7. REFERENCES

[1] P. Cano, A. Loscos and J. Bonada. Score-Performance Matching using HMMs. In *Proceedings of the International Computer Music Conference*, pp. 441-444 1999.

[2] A. Cont. Realtime Audio to Score Alignment for Polyphonic Music Instruments Using Sparse Non-Negative Constraints and Hierarchical HMMs. In *IEEE International Conference in Acoustics and Speech Signal Processing*, pp. V245–V248, 2006.

[3] A. Cont. Modeling Musical Anticipation: From the Time of Music to the Music of Time. PhD. thesis, 2008.

[4] N. Montecchio and N. Orio. Automatic Alignment of Music Performances with Scores Aimed at Educational Applications. In *Proceedings of the International Conference on Automated solutions for Cross Media Content and Multi-channel Distribution*, pp. 17–24, 2008.

[5] N. Orio and F. Déchelle. Score Following Using Spectral Analysis and Hidden Markov Models. In *Proceedings of the International Computer Music Conference (ICMC)*, pp. 125-129, 2001.

[6] L.R. Rabiner. A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition. In *Proceedings of the IEEE*, 77(2):257-286, 1989.

[7] C. Raphael. Automatic Segmentation of Acoustic Musical Signals Using Hidden Markov Models. In *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 21(4):360–370, 1999.

[8] C. Raphael. Aligning Music Audio with Symbolic Scores using a Hybrid Graphical Model. *Machine Learning*, 65:2(389–409), 2006.