Score Following Using
Spectral Analysis and Hidden Markov Models

Nicola Orio, François Déchelle
Ircam – Centre Georges-Pompidou
email: {norio,dechelle}@ircam.fr

Abstract

This paper presents an approach to score following. The real-time alignment of a performance with a score is obtained through the use of a hidden Markov model. The model works on two levels. The lower level compares the features of the incoming signal with the expected ones. Groups of states of the lower level are embedded in states at the higher level, which are used to model the performance by taking into account the possible errors a performer may make. The performer's position on the score is computed through a decoding technique alternative to classic Viterbi decoding. A novel technique for the training of hidden Markov models is proposed.

1 Introduction

Electroacoustic music often requires the synchronization between musicians and algorithms that synthesize sound. The simplest solution to the synchronization problem, which has its roots in tape music, is to ask musicians to follow the synthetic performance. This solution, which Vercoe defined as the "music-minus-one syndrome" (Vercoe 1984), is highly demanding for musicians, who have to play a difficult piece, while looking at the score and at a timer at the same time, and who are not allowed to make errors. Moreover, expressive changes of tempo, which is likely to vary in different performances, are not possible. Another solution is to ask a technician to control the sound synthesis parameters in real-time. Given the complexity of contemporary pieces, this task will require a well-trained musician, able to follow the score and interact on a computer interface and recognize possible musician's errors. The role of hidden performer is not particularly satisfying and prone to imprecision.

The problem of real-time automatic synchronization among musicians and computers, which is called *score following*, has been investigated since 1984, when the first two papers appeared. This paper presents an approach to score following based on the use of a two-level Hidden Markov Model (HMM). Alignment is computed through decoding. A novel technique for the training of the parameters is proposed.

2 Background

The problem of matching a performance with a score can be considered a special case of sequence alignment, which has been extensively addressed in other research areas, notably in speech recognition and in molecular genetics. In both these domains, HMMs have become extremely popular due their outstanding results. Moreover, they are applied to all domains (e.g., hand-gesture recognition, fault-tolerance) where it is possible to take advantage of a trainable model of the process that is analyzed. Applying HMMs to score following seems then promising.

2.1 Overview on Hidden Markov Models

HMMs are probabilistic finite-state automata, where transitions between states are ruled by probability functions. At each transition, the new state emits a value with a given probability. Emissions can be both symbols from a finite alphabet and continuous multidimensional values. Transition probabilities are assumed to depend only on a finite number of previous transitions (usually one) and they may be modeled as a Markov chain. The presence of transitions with probability equal to zero defines a topology of the model, limiting the number of possible paths.

States $Q = \{q_1, q_2, \ldots, q_N\}$ are not observable, what can be observed are only their emissions $O_T = \{o_1, o_2, \ldots, o_T\}$ from time 1 to time T, which are called observations. The problem of finding, given a sequence of observations O_T, which is the optimal (in some sense) corresponding sequence of states $q_1, \ldots, q_T$ is called *decoding*. The set of parameters λ of a HMM, namely transition and emission probabilities, can be *trained* to maximize the probability of emitting a given set of observation sequences. A complete discussion on theory and applications of HMMs can be found in (Durbin et al. 1998) and in (Rabiner and Juang 1993).

2.2 Early Approaches to Score Following

There are two main approaches to score following in the literature. They may be defined the *note* and the *signal* ap-

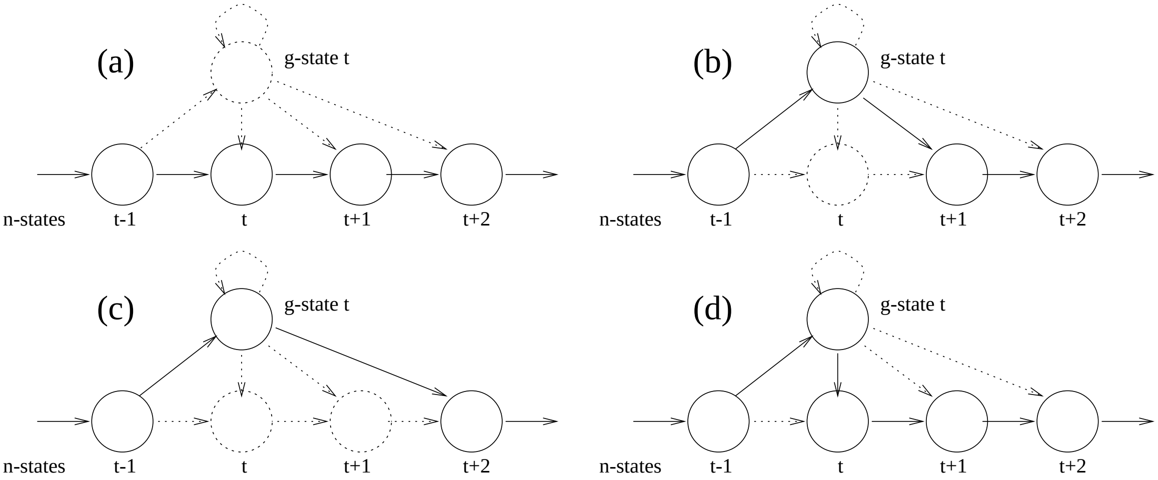

Figure 1: Graphical representation of possible path between states at the higher level (only a g-state is shown for simplicity); solid paths are related to the presence of a *correct* (a), *wrong* (b), *skip* (c), and *extra* (d) note at time t.

proaches. The former considers performer's error as the cause of mismatch, while assuming a reliable input from a MIDI instrument; the latter focuses on errors due to an incorrect detection of sound features, while not explicitly dealing with musician's errors.

Algorithms concerned with the note approach are usually based on string matching techniques with some added heuristics to prevent errors in real-time decisions, as in the early works by Dannenberg and Mukaino (1988) and Vercoe (1984). A simpler technique (Puckette 1990) is to compare the incoming events with the list of the expected events and choose the first exact match as the candidate for the alignment. Puckette (1995) proposed to merge the note and the signal approaches by introducing two different pitch trackers (one fast and imprecise and the other slower but more reliable), in order to deal with the imprecisions in note detection.

2.3 Related Works

Some approaches to score following using statistical tools have been presented in the literature. Grubb and Dannenberg (1998) proposed a method for calculating the position of a performer on the score based on a probability density function (pdf) conditioned on both the estimated distance with the previous position and the most recent observations. Observations are multidimensional features, including fundamental pitch, spectral envelope, and amplitude changes. Cano, Loscos, and Bonada (1999) used an HMM where the emissions were a number of relevant sound features, like energy, zero crossing, fundamental frequency, with their derivatives. The HMM is left-to-right, with self-transitions modeling note length. The alignment is computed through Viterbi decoding. Both approaches have the disadvantage of relying on the robustness of algorithms of pitch tracking.

The approach proposed by Raphael (1999) is strictly related to the present work. The alignment is computed through the use of a HMM. While possible errors made by performers are not explicitly considered, there is no dependence on pitch tracking routines because states directly emit spectral features. Note durations are modeled in two different ways, depending on their length. A decoding technique alternative to Viterbi is used for the alignment, while the training is performed using classical Baum-Welch algorithm.

3 A Novel Score Follower

The approach described in this paper merges the note and the signal approaches by explicitly taking into account performer's errors while using the audio signal as the input. A two-level HMM is used for separately modeling the performance as a sequence of musical events and the signal as a sequence of features. Real-time synchronization is carried out through a decoding technique suitable for local alignment. A new technique is proposed for the training of the HMM.

3.1 Modeling the Performance

States at the *higher level* model the events written in the score, together with the possible errors that the performer can make. Events may be rests, notes, trills, chords, and so on. There are two categories of states: *normal states* (n-states), which correspond to events correctly played, and *ghost states* (g-states), which correspond to a local mismatch between an event in the score and the actual performance. Each event is represented by a n-state and a parallel g-state.

The topology of the HMM is left-to-right, in accordance with the temporal precedence of score events. Each n-state is connected to the subsequent n-state and g-state. Transitions from g-states take into account the three classes of possible errors: wrong, extra, and skip notes. As it can be seen from Figure 1, different paths in the HMM graph correspond to different errors.

It may be argued that, even without an explicit care for performer's errors, the HMM should be robust enough to find the correct alignment after an error occurred. A number of tests using HMM with only n-states, showed that the proposed model is faster in finding the correct alignment after some errors. Moreover, the use of g-states allows the system to know when an error has been made and thus alternative actions can be programmed.

3.2 Modeling the Signal

States at the *lower level* model the incoming signal. Each state at the higher level is made by a set of states at the lower level. These states take into account, for each score event, the features related to the attack, the sustain, and the possible silence at the end. Figure 2 shows how states at the lower level are connected to form a single state at the higher level.

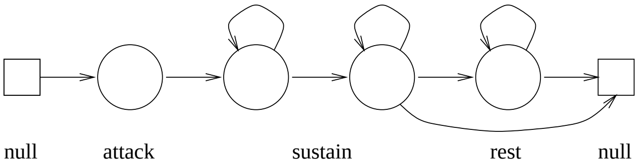

Figure 2: Graphical representation states at the lower level embedded in a note event of the higher level (two null states are added).

As can be seen in Figure 2, event duration is modeled using a cluster of sustain states with non-null self-transition probability. Given n the number of states in a cluster and p their self-transition probability, the probability of having a given duration has the form of the negative binomial law:

$$P(d) = \left(\begin{array}{c} d-1 \\ n-1 \end{array} \right) p^{d-n}(1-p)^n \qquad (1)$$

The values n and p can be choose for setting the position of the maximum and the shape of the curve, which are related to the expected duration and to the precision in time resolution.

The emissions of the states at the lower level are related to the probability of emitting a given spectrum. In particular, it is proposed to model the probability that the energy carried by the first partials is a given percentage of the overall signal energy. This can be performed by computing the energy output of a bank of bandpass filters, centered on the harmonic sequence of the expected note, divided by the global energy of the signal. The parameters to be set are the number of bandpass filters, their shape and bandwidth. A description of the filtering step is given in (Orio and Schwarz 2001).

Together with this feature, the log-energy of the signal is used. Moreover, their derivatives are computed to better deal with signal attack. Finally, the ratio between odd and even harmonics is computed to prevent from doubling of octaves. Once these parameters are set, emission probabilities can be trained as discussed in Section 3.4.

The approach allows for synchronization with polyphonic signals, with the only difference that the filterbank is set according to the superposition of the different notes in the polyphony. Moreover, it is possible to model the spectral features of more complex musical gestures, as trills and frequency vibrato.

3.3 Decoding for the Alignment

Real-time alignment can be carried out through decoding. Viterbi algorithm, which is a technique widely used, finds the state sequence $\{q_1, \ldots, q_T\}$ that most likely generated the complete sequence of observations $\mathbf{O_T}$. This criterion is hence of global optimality, which may not be the best choice for score following. In fact, for real-time alignment the main goal is finding the state locally corresponding to the actual note. This is a particular case of another optimality criterion, already introduced in molecular genetics (Durbin et al. 1998), where the probability of local alignments of each individual state is maximized.

This criterion may be applied by considering the optimal alignment of only the last state, that is the one corresponding to the current position in the score. The decoding of the last state at time t is performed through the following maximization:

$$q_t = \arg\max P(q_t = v_t, o_1, \ldots, o_t \mid \lambda) \qquad (2)$$

The function to be maximized corresponds to the *forward variables* (Rabiner and Juang 1993). Like Viterbi, also this decoding can then be computed using a dynamic programming approach at a low computational cost. During the online computation of the forward variables, scaling can be applied to prevent values to exceed the precision range of the machine. The comparison of this decoding with Viterbi showed lower delay in detecting note changes and higher robustness to errors.

3.4 Training the HMM parameters

Training presents some difficulties. Even if rehearsals can be used to collect sample data, there is a risk of model overfitting. For instance, a transition towards a g-state that corresponds to an error that never occurred during rehearsal may become highly unlikely, with a decrease in robustness to new errors. Moreover, training should be performed *before* the following of a new piece, to allow for synchronization at first rehearsal. That is particularly suitable with new productions when the score is likely to change between different rehearsals. It is proposed to carry out a first training of the HMM by using a database of sounds to train the emissions of the lower level, and a set of automatic generated performances to train the transitions at the higher level.

Samples from a web database (Studio-Online 2001) have been used to compute *emission probabilities*. The database contains samples of all the orchestral instruments, played with different techniques and with different dynamics. The continuous multidimensional features of the signal, described in Section 3.2, are assumed to be statistically independent. After a number of tests, the exponential pdf has been chosen to model each of the features. States clustered in a n-state are trained by analyzing the samples with their correct filterbank. States clustered in a g-state are trained by using the filterbank of the parallel n-state but analyzing samples *not corresponding* to the expected event; that is, if a filterbank is set for tone A4, the output is computed when all tones but A4 are filtered.

Probabilities of self-transitions at the lower level are set according to events durations (see Section 3.2). Hence, only *transition probabilities* between states at the higher level need to be trained. To this end, a number of performances is automatically created and used as examples for the training. They include correct performances and performances with all the possible errors affecting up to four subsequent events.

Normally, training of HMM is performed using the Baum-Welch method that maximizes the probability that the model will emit a given set of sequences of observations. Experiments with this training showed that the procedure is not suitable for the proposed topology, because during the training the meaning of the states (n-states vs. g-states) may be lost. Performer's errors could not be recognized by the system, and no alternative action can be set (i.e., avoiding changes in local tempo or no triggering of events).

A novel technique has been developed for the training of HMMs. The probability of being in the correct n-state or g-state is maximized, given a sequence of observations. Hence, instead of maximizing $P(\mathbf{O_T} \mid \lambda)$, as in Baum-Welch technique, the following quantity is maximized:

$$\prod_{k=1}^{K} \prod_{t=1}^{T^{(K)}} P(q_t = v_t^{(k)}, o_1^{(k)}, \ldots, o_t^{(k)} \mid \lambda) \qquad (3)$$

where, for each performance k, $T^{(K)}$ is the total duration of the performance, $v_t^{(k)}$ is the correct last state, $o_1^{(k)}, \ldots, o_t^{(k)}$ are the observation until time t. Performances include both the ones automatically generated and the ones recorded during rehearsals. The only difference is that, for the latter, the user has to specify the correct last state in case of mismatch of the follower.

It can be seen that this technique is coherent with Equation 2 used for the decoding. The calculation can be performed similarly to the Baum-Welch algorithm, with the only difference that the last state has to be explicitly included in the calculation of the Expectation step.

4 Results and Future Work

The proposed approach to score following has been successfully tested on a number of performances of contemporary music, which are usually more difficult to follow, using different acoustic instruments and the voice. Tests have been also developed using polyphonic performances and performances where trills, vibrato, and sharp staccato were present, with good results. Robustness to performers errors has been tested by using the same performance and altering the score. This allowed to test the methodology, even if a database of real performances with errors will be more appropriate.

In the future, extensive tests with music students, who may be more likely to make errors, will be carried out. The methodology will be extended for dealing with other sound events where pitch is irrelevant, like noises and percussive sounds. This extension will affect the lower level, by introducing new kinds of emission probabilities that better describes these sound events.

References

Cano, P., A. Loscos, and J. Bonada (1999). Score-performance matching using hmms. In *Proc. of ICMC*, pp. 441–444. ICMA.

Dannenberg, R. B. and H. Mukaino (1988). New techniques for enhanced quality of computer accompaniment. In *Proc. of ICMC*, pp. 243–249. ICMA.

Durbin, R., S. Eddy, A. Krogh, and G. Mitchison (1998). *Biological sequence analysis*. Cambridge, UK: Cambridge University Press.

Grubb, L. and R. B. Dannenberg (1998). Enhanced vocal performance tracking using multiple information sources. In *Proc. of ICMC*, pp. 37–44. ICMA.

Orio, N. and D. Schwarz (2001). Alignment of monophonic and polyphonic music to a score. In *Proc. of ICMC*. ICMA.

Puckette, M. (1990). Explode: A user interface for sequencing and score following. In *Proc. of ICMC*, pp. 259–261. ICMA.

Puckette, M. (1995). Score following using the sung voice. In *Proc. of ICMC*, pp. 175–178. ICMA.

Rabiner, L. and B. Juang (1993). *Fundamentals of Speech Recognition*. Englewood Cliffs, NJ: Prentice-Hall.

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.

Studio-Online (July 20th, 2001). Server of sound and sound-processing. `http://sol.ircam.fr`.

Vercoe, B. (1984). The synthetic performer in the context of live performance. In *Proc. of ICMC*, pp. 199–200. ICMA.