Score Following at Ircam

Arshia Cont
Ircam / UCSD
1 Place Igor Stravinsky
75004 Paris, FRANCE.
cont@ircam.fr

Diemo Schwarz
Ircam, Realtime Applications Team
1 Place Igor Stravinsky
75004 Paris, FRANCE
schwarz@ircam.fr

Abstract

This paper describes the submission to the MIREX'06 (Music Information Retrival Evaluation eXchange) first score following tasks.

1. Overview

Score following is the key to an interaction with a written score/song based on the metaphor of a performer with an accompanist or band. For a historical review of score follower systems we refer the reader to [1, 2].

The Score Following Player submitted accepts monophonic audio and MIDI input from the performer. The audio following modules uses a core algorithm based on Hidden Markov Models (HMM). Figure 1 shows a block diagram overview of the follower algorithm.

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 to their outstanding results. The HMM in Figure 1 can also be viewed as a sequential model of the *score* where the states (score events) can not be directly observed. What is observed by the system is the probabilities assigned to each state of the score model which are used consequently by a decoding algorithm to match the realtime audio to an event in the score. In the following section we describe the methodology for each block in Figure 1.

2. Score model

The music score is modeled as HMMs where each state represents an event in the score. The topology of the HMM is left-to-right, in accordance with the temporal precedence of score events. Each event in the score is modeled as a sequence of states. 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 a sample

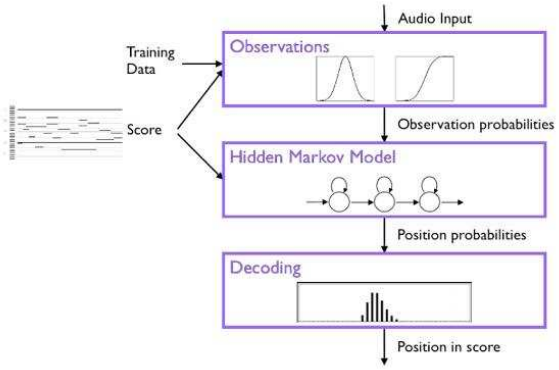

Figure 1. Overview of Score Follower system

note model used in our system. A *silence* model is essentially the same but with *rest* states instead of *sustains* and *attacks*.

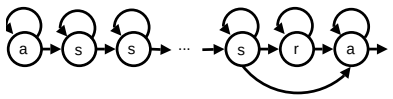

Figure 2. Note model used in Score HMM. `a` stands for *attack*, `s` for *sustain* and `r` for *rest*.

The number of states (n) and transition probabilities (p for forward and $1-p$ for self) at each low level event is determined by solving for n and p in a binomial distribution given the mean (np) as the duration of the event according to the given score and a fixed variance ($np(1-p)$) (50% of the duration here).

Given this note model in terms of HMMs, a score can be represented by accumulating all the note and rest events according to the score and in a sequential manner. Figure 2 shows a sample score and its corresponding score model.

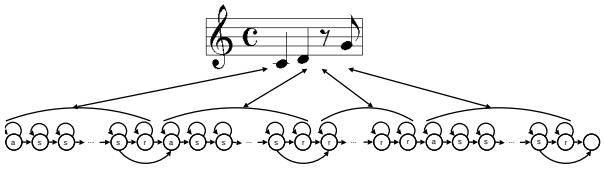

Figure 3. Sample score and corresponding HMM model.

3. Observation Modeling

Observation in the context of our system consists of calculating features from the audio spectrum in real-time and associate the desired probabilities for low-level HMM states. Low-level states in our system are *attack, sustain* and *rest* for each note in the score. Spectrum features are Log of Energy, Spectral Balance and Peak Structure Match (PSM). We will not go into implementation details of the mentioned features which are described in [1, 3, 2]. The observation process can be seen as a dimension reduction process where a frame of our data, or the FFT points, lies in a high dimensional space $\Re^J$ where J is 2048. In this way, we can consider the features as vector valued functions, mapping the high dimensional space into a much lower dimensional space, or more precisely to $2 + N$ dimensions where N is the number of different notes present in the score for the PSM feature. Another way to look at the observation process is to consider it as a probability mapping between the feature values and low-level state probabilities. A diagram of the observation process is demonstrated in Figure 4. In this model, we calculate the low-level feature probabilities associated with each feature which in terms are multiplied to obtain a certain low-level state feature probability. As an example, the Log of Energy feature will give three probabilities Log of Energy for Attack, Log of Energy for Sustain and Log of Energy for Rests. In order to calculate probabilities from features, each of the 8 low-level state feature probabilities is using probability mapping functions from a database of stored trained parameters. They are derived from Gaussians in forms of cumulative distribution functions (CDFs), inverse cumulative distribution functions or PDFs depending on the heuristics associated with each feature state. Note that the dimension of each model used is one at this time. By this modeling we have assumed that the low-level states' attributes are global which is not totally true and would probably fail in extreme cases. However, due to a probabilistic approach, training the parameters over these cases would solve the problem in most cases we have encountered. Another assumption made is the conditional independence among the features, responsible for the final multiplication of the feature as in Figure 4.

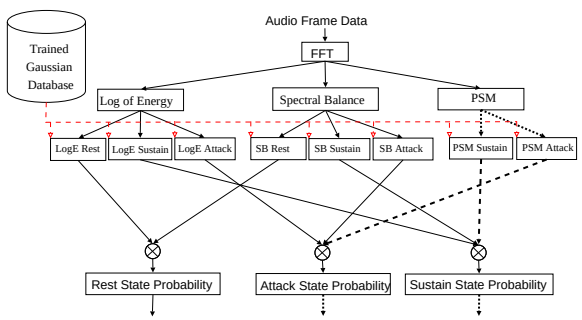

Figure 4. Probability Observation Diagram

4. Decoding and Alignment

Once the observation probabilities are calculated for states in the score model, they are used by a decoding scheme to decide what is the appropriate current high-level state using present and past information. The Bayesian framework in this submission considers observation probabilities as $P(y_t|x_t^k)$ where y_t is a realtime audio observation at time t and x_t^k would be the (hidden) state k in the score. Using this scheme, the current *belief* of the system is computed as in Equation 1 where Z is a normalizing constant, $P(x_t^k|x_{t-1}^{k^*})$ is the transition prior from the score model and $P(x_{t-1}^k|y_{1:t-1})$ is the previous belief of the system. This way, the current high-level state can be decoded by Equation 2.

$$P(x_t^k|y_{1:t}) = \frac{1}{Z} P(y_t|x_t^k) P(x_t^k|x_{t-1}^{k^*}) P(x_{t-1}^k|y_{1:t-1}) \quad (1)$$

$$k^* = \underset{k}{\operatorname{argmax}}\, P(x_t^k|y_{1:t}) \quad (2)$$

5. Conclusion

In this paper, we briefly described our submission to MIREX'06 Score Following task. In a real-world application, our observation model accepts piece-specific and instrument-specific trained data as parameters of each CDF described above. For this submission, since training was not considered for the contest, we use default parameters. Extensions to the system described in this paper as well as more material can be found in [4] and references herein.

References

[1] Arshia Cont, "Improvement of observation modeling for score following," Dea atiam, University of Paris 6, IRCAM, Paris, 2004.

[2] Nicola Orio, Serge Lemouton, Diemo Schwarz, and Norbert Schnell, "Score Following: State of the Art and New Developments," in *Proceedings of the International Conference on New Interfaces for Musical Expression (NIME)*, Montreal, Canada, May 2003.

[3] Nicola Orio and Diemo Schwarz, "Alignment of Monophonic and Polypophonic Music to a Score," in *Proceedings of the ICMC*, Havana, Cuba, 2001.

[4] Arshia Cont, "Realtime audio to score alignment for polyphonic music instruments using sparse non-negative constraints and hierarchical hmms," in *IEEE ICASSP*. May 2006, Toulouse.