Training IRCAM's Score Follower

Arshia Cont
IRCAM - Centre Pompidou
Realtime Applications Group
1 pl. Stravinsky, Paris 75004.
acont@ircam.fr

Diemo Schwarz
IRCAM - Centre Pompidou
Realtime Applications Group
1 pl. Stravinsky, Paris 75004.
schwarz@ircam.fr

Norbert Schnell
IRCAM - Centre Pompidou
Realtime Applications Group
1 pl. Stravinsky, Paris 75004.
schnell@ircam.fr

Abstract

This paper describes our attempt to make the *Hidden Markov Model (HMM)* score following system developed at IRCAM sensible to past experiences in order to adapt itself to a certain style of performance of musicians on a particular piece. We focus mostly on the aspects of the implemented machine learning technic pertaining to the style of performance of the score follower. To this end, a new observation modeling based on Gaussian Mixture Models is developed which is trainable using a novel learning algorithm we would call *automatic discriminative training*. The novelty of this system lies in the fact that this method, unlike classical methods for *HMM* training, is not concerned with modeling the music signal but with correctly choosing the sequence of music events that was performed.

Introduction

The subject of score following has been studied for almost 20 years now. The goal is to simulate the behavior of a musician playing with another, a "synthetic performer", to create a virtual accompanist that follows the score of the human musician. For an introduction and state of the art on the topic of score following and details of the IRCAM system, we refer the curious reader to (Orio *et al.* 2003) and Chapter 1 in (Cont 2004). In this paper, we introduce the learning algorithm used for IRCAM's score follower with a focus on learning music performance style.

We begin this paper by a review of past attempts in score following literature, focusing on the adaptability and learning aspects of the algorithms. This section is followed by an overview of our approach and objective towards training leading to a new *observation modeling* for score following. This new design can articulate specific behavior of the musician in a controllable manner. In this respect, the system would be able to grab "stylistic" behavior of different musicians on the same score and on low-level musical features such as attacks, note sustains and releases.

After reviewing the proposed architecture and based on the same approach, we introduce a learning algorithm called *automatic discriminative training* which conforms to the practical criteria of a score following system. The novelty of this system lies in the fact that this method, unlike classical methods for *HMM* training, is not concerned with modeling the music signal but with correctly choosing the sequence of music events that was performed. In this manner, using a *discrimination* process we attempt to model class boundaries rather than constructing an accurate model for each class.

Finally, we demonstrate the results on contemporary music repertoire such as pieces by Philippe Manoury and Pierre Boulez, and in the case of the former using live musicians focusing on how the system discriminates between different interpretations of the same piece performed by different musicians.

Background

The first debate on learning in the context of score following occurred in Vercoe and Puckette's historical score following in (Vercoe & Puckette 1985). In describing the objective of training the score follower, we quote from the original article:

> . . . [speaking about the 1984 score follower] there was no performance "memory", and no facility for the synthetic performer to learn from past experience. . . since many contemporary scores are only weakly structured (e.g. unmetered, or multi-branching with free decision), it has also meant development of score following and learning methods that are not necessarily dependent on structure (Vercoe & Puckette 1985).

Their learning method, interestingly statistical, allows the synthetic performer to rehearse a work with the live performer and thus provide an effective performance, called "post-performance memory messaging." This non-realtime program begins by calculating the mean of all onset detections, and subsequently tempo matching the mean-corrected deviations to the original score. The standard deviation of the original onset regularities is then computed and used to weaken the importance of each performed event. When subsequent rehearsal takes place, the system uses these weighted values to influence the computation of its least-square fit for metrical prediction.

While in Roger Dannenberg's works before 1997 (or more precisely before the move to a statistical system) there is no report of an explicit training, in Puckette's 95 article there are evidences of off-line parameter control in three instances: defining the weights used on each constant-Q filter

associated with a partial of a pitch in the score, the curve-fitting procedure used to obtain a sharper estimate of f_0 and threshold used for the input level of the sung voice. According to (Puckette 1995), he did not envision any learning methods to obtain the mentioned parameters. In the first two instances he uses trial and error to obtain global parameters satisfying desired behavior and the threshold is set by hand during performance. Note that in different performances of the same piece, it is this hand-setting of parameters which correlates to the performance style of the musician.

By moving to the probabilistic or statistical score followers, the concept of training becomes more inherent. In Dannenberg and Grubb's score follower (Grubb & Dannenberg 1997), the probability density functions (PDFs) should be obtained in advance and are good candidates for an automatic learning algorithm. In their article, they report three different PDFs in use and they define three alternative methods to obtain them:

> First, one can simply rely on intuition and experience regarding vocal performances and estimate a density function that seems reasonable. Alternatively, one can conduct empirical investigations of actual vocal performances to obtain numerical estimates of these densities. Pursuing this, one might actually attempt to model such data as continuous density functions whose parameters vary according to the conditioning variables (Grubb & Dannenberg 1997).

Their approach for training the system is a compromise of the three mentioned above. A total of 20 recorded performances were used and their pitch detected and *hand-parsed* time alignment is used to provide an observation distribution for actual pith given a scored pitch and the required *PDF*s would be calculated from these hand-discriminated data.

In the *HMM* score following system of Raphael (1999), he trains his statistics (or features in our system's terminology) using a *posterior marginal distribution* $\{p(x_k|\mathbf{y})\}$ to re-estimate his feature probabilities in an iterative manner (Raphael 1999). In his iterative training he uses *signatures* assigned to each frame for discrimination but no parsing is applied beforehand. In his latest system, incorporating *Bayesian Belief Networks (BBN)*, since the *BBN* handles temporal aspect of the interpretation, several rehearsal run-throughs are used to compute the means and variances of each event in the score, specific to that interpretation (Raphael 2001).

In the case of Pardo and University of Michigan's score follower, a training is done to obtain the *probabilistic costs* which is independent of the score and performance and is obtained by giving the system some musical patterns such as arpeggios and chromatic scales (Pardo & Birmingham 2002). In this manner, the system should be incapable of considering performance style issues.

Approach

Training in the context of score following is to adapt its parameters to a certain style of performance and a certain piece of music. On the other hand, in the context of musical production at IRCAM and working with different musicians and composers, implementation of a training system should ideally be unsupervised or at least automatic, without adding anything to the tradition of music performance. In other words, the system is created to realize the music as opposed to selecting music to demonstrate the technology.

In this respect, we envision a system which learns or adapt itself through a database of sound files of previous performances of the same piece or in the case of a creation, of recorded rehearsals. After the offline and automatic learning, the system is adapted to a certain style of performance, and thus provides better synchronization with the score in real-time.

Figure 1 shows a general diagram of *IRCAM*'s current score follower as a refinement of the model described in (Orio & Déchelle 2001). In our approach to the described problem, we have refined the *observation modeling* (the top block in diagram) in order to obtain the desired architecture. The *decision and alignment block* (lower block in diagram) is described in details in (Orio & Déchelle 2001).

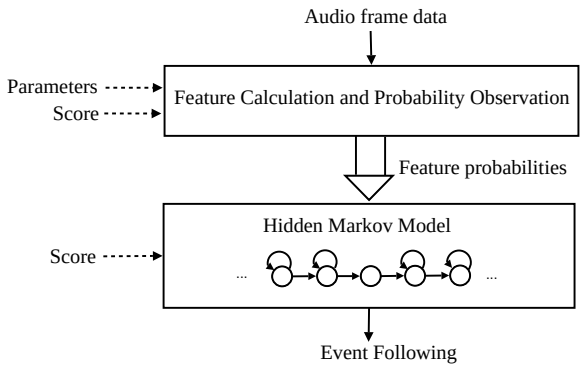

Figure 1: General diagram of *IRCAM*'s score following

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 states. Low-level states used in our system are *attack*, *sustain* and *release* for each note in the score; and spectrum features used are *Log of Energy*, *Spectral Balance* and *Peak Structure Match*. We will not get into the implementation details of the features for two reasons:

- The proposed algorithm and design is independent of the features and acts directly on the output of the features disregarding their implementation details.
- These details are covered in (Orio & Schwarz 2001; Cont 2004) and are not of the main interest of this paper.

It suffices to know that the *Spectral Balance* feature gives a measure of balance between low-frequency and high-frequency contents of an analysis time-frame and the *Peak Structure Match (PSM)* provides a measure of the spectral pitch for every note in the score at each time-frame.

The observation process can be seen as a dimension reduction process where a frame of our data, or the FFT points,

lie in a high dimensional space, $\Re^J$ where J=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.

In our model, we calculate the low-level state probabilities associated with each feature which in terms would be multiplied to obtain a certain low-level state 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 Release*. A diagram of the observation process is demonstrated in Figure 2.

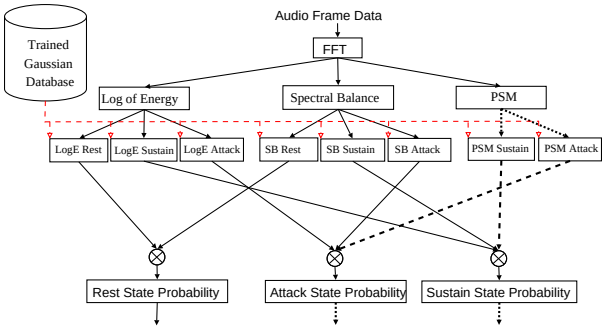

Figure 2: Probability Observation Diagram

In order to calculate probabilities, each low-level state feature probability (third layer in Figure 2) is using probability mapping functions from a database with stores trained parameters. The probability mapping is derived from gaussians in forms of cumulative distribution function (CDF), inverse cumulative distribution function or a PDF depending on the heuristics associated with each feature state. This architecture is inspired by Gaussian Mixture Models. Note that the dimension of each model used is one at this time.

By this modeling we have assumed that low-level states' attributes are not local 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 2.

Note that the observation process is in real-time and during the score alignment.

Training the Score Follower

Training is to adapt the *observation* parameters to a certain piece and certain style of performance. Speaking about training for score following often initiates fear of system obsolesce and portability for musicians and composers using the system. For this reason, we tend to specify what we mean exactly by training.

In an ideal training, the system runs on a huge database of *aligned* sound files and adapts its parameters to the performance. In this case, the training is usually supervised and is integrated in the system's practice. However, in a musical situation dealing with traditions of music rehearsals and performances,

- Musicians prefer no additional item added to their practice situation.
- No database of *aligned* audio exists and moreover, working in the context of contemporary music limits the availability of different performances and recordings for a piece.
- Whatever added to the system in general, should not reduce the portability of the piece. Musicians travel with the piece!

The above constraints would limit the ideal training to an *unsupervised* training, having few or just one rehearsal runthroughs to be observed. In this context, the training will be off-line and would use the data during rehearsal to train itself. Atleast for portability issues, training should be *automatic*.

Also from a developmental point of view, since score following is a *work in progress* as composers' demands increase and change, training should be ideally independent of features so that by introducing new features, training does not need any change.

In this manner, with an ideal learning algorithm the system should be capable of modeling different styles of performances of a certain piece, giving more accurate models for audio to score alignment.

The automatic discriminative training

In score following we are not concerned with estimating the joint density of the music data, but are interested in the posterior probability of a musical sequence using the acoustic data. More informally, we are not finally concerned with modeling the music signal, but with correctly choosing the sequence of music events that was performed. Translating this concern to a local level, rather than constructing the set of *PDFs* that best describe the data, we are interested in ensuring that the correct *HMM* state is the most probable (according to the model) for each frame.

This leads us to a *discriminative training* criterion. This criterion has been described in (Renals *et al.* 1993) and in the context of neural networks and not *HMM* models. Discriminative training attempts to model the class boundaries — learn the distinction between classes — rather than construct as accurate a model as possible for each class. In practice this results in an algorithm that minimizes the likelihood of incorrect, competing models and maximizes the likelihood of the correct model.

While most discriminative training methods are supervised, for portability issues and other reasons discussed before, we need our training to be automatic if not unsupervised. For this reason, we introduce an automatic supervision over training by constructing a *discrimination knowledge* by an alternative algorithm which forces each model to its boundaries and discriminates feature observations. *Yin*

(de Cheveigne & Kawahara 2002) has been chosen as this algorithm to provide discrimination knowledge.

Figure 3 shows a diagram of different steps of this training. The inputs of this training are an audio file plus its score. There are two main cores to this system: *Discrimination* and *Training*.

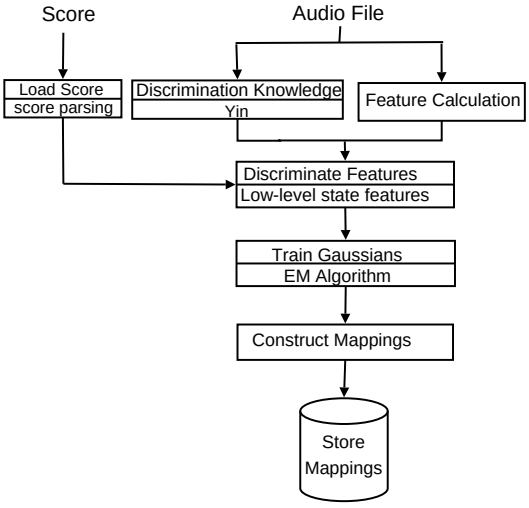

Figure 3: Automatic Discriminative Training Diagram

Discrimination

By discrimination, we aim to distinguish low-level states in the feature domain. In this process, as part of the training, a set of states and their corresponding observations would be obtained without actually segmenting or labeling the performance. The *Yin* algorithm (de Cheveigne & Kawahara 2002) is used as the base knowledge. Note that *Yin* is originally a fundamental frequency estimator and provides fairly good measures of aperiodicity of each analysis frame. By a one-to-one correspondence between the observed data frames and *Yin*'s analysis frames, and using *Yin*'s information for each frame we decide on the type of the associated low-level state (*Attack*, *sustain* and *release*) for each note in the score.

Figure 4 shows *Yin*'s f_0 output together with score information as bands for each different note in the score (Manoury's "En Echo" in this case), used to find event indices in analysis. The aperiodicity measure for each frame discriminates between *release* and *note* events and if the detected note meets a minimum time length, about 20 frames around the first index would be marked as the *attack* frame indices as well as the rest for *sustain* frames. Using these indices, we discriminate *attack*, *release* and *sustain* frames from each feature's observation. Obviously, each observation frame is assumed to have only one state type.

Figure 5 shows Log of Energy feature histograms on a particular piece ("En Echo" by Philippe Manoury) along with histogram for discriminated states in the same observation.

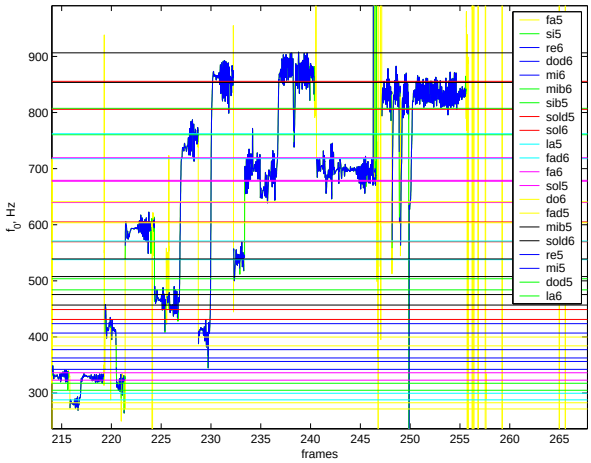

Figure 4: Discrimination using *Yin*

Training

Having all features discriminated, we are ready to train the *gaussians*. We evade using fitting algorithms due to robustness and since we are dealing with *gaussian mixtures* (Reynolds 1995) and are planning more mixtures in a row for future, we use *EM Algorithm* (Bilmes 1998; Dempster, Laird, & Rubin 1977) to construct the *gaussians* on observed discriminated features.

The result of the training is a set of *PDF*s that correspond to each low-level state feature. We go further and construct structures containing μ and σ values for each *PDF* as well as the corresponding type of *probability mapping* for each state feature and probability range and observed feature's range for calibration. This way each file structure would correspond to one state feature with all the above information. This data will be stored in a database which will be used in the real-time score follower's *observation* block as shown in Figure 2.

Response to performance style

Having the system trained and tested on different performances of the same piece with different musicians, the system tends to respond better in critical situations which are mostly due to the style of performance and adaptability of the new system. This becomes more clear when using one performer's parameters on another and on the same piece, which would lead to imprecise alignment on critical phrases. This is mainly due to the fact that there are eight different gaussians trained on different attributes of acoustic data and mostly thanks to the *discriminative training* which emphasizes the distinction between each class rather than constructing an accurate model.

Figure 6 compares the gaussians obtained for *Log of Energy* on three different performances of "En Echo" for voice and computer of the composer Philippe Manoury, performed by Michel-Dansac Donatienne, Valerie Philippin and Francoise Kubler in different sessions as an example.

Note that there are major differences for the release states of the two performances. This is due to the fact that release

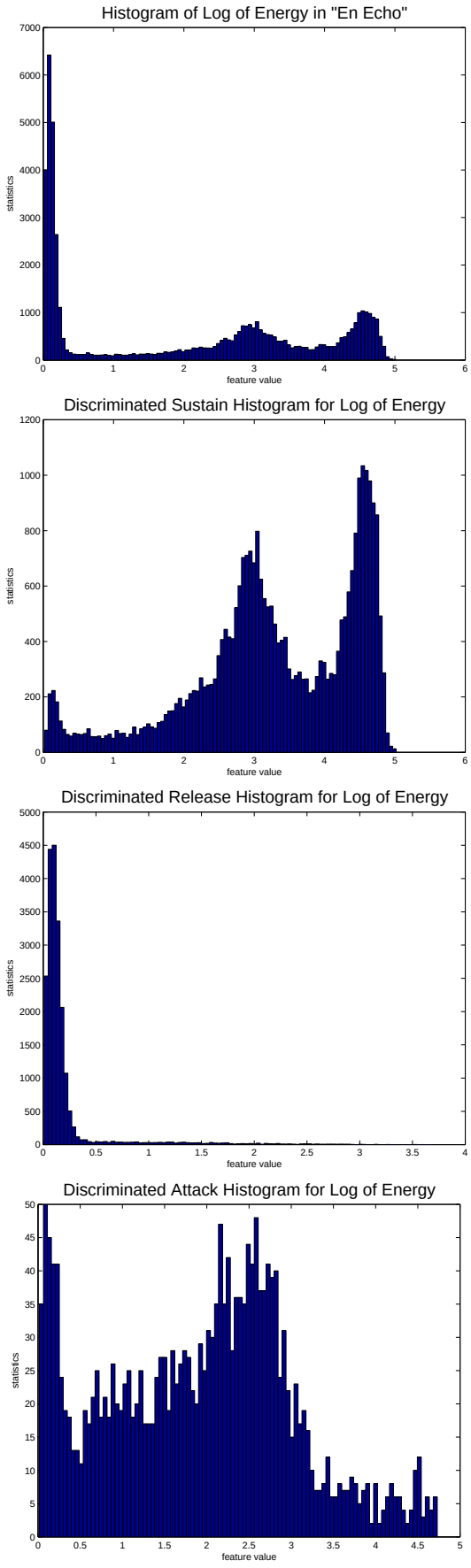

Figure 5: Discrimination for LogE feature in ”En Echo” with Donatienne as Soprano - From top to bottom: Log of Energy histogram (non-discriminated), Discriminated Sustain Log of Energy, Discriminated Release Log of Energy, Discriminated Attack Log of Energy.

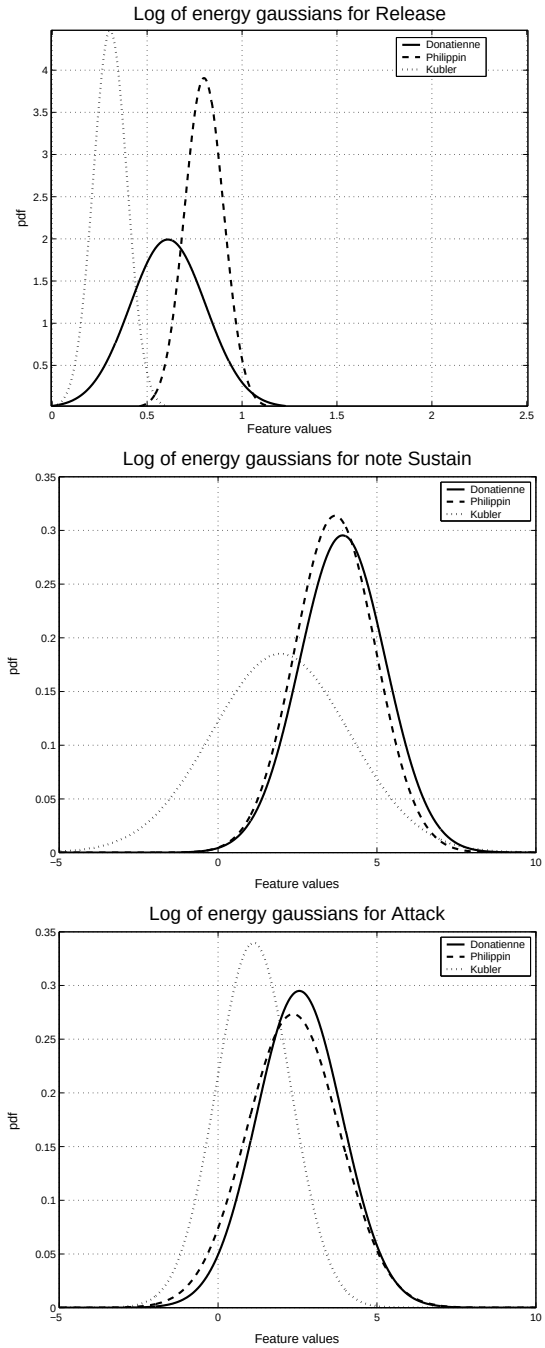

Figure 6: Trained gaussian examples for two separate performances of ”En Echo” of Manoury for soprano and electronics, performed by Michel-Dansac Donatienne, Valerie Philippin and Francoise Kubler.

or silence correspond to background noise which is different for the two performances. For the *Log of Energy* feature, it mainly acts as a threshold and during a live performance it would be a matter of calibration of feature output to the range observed during training. However, in the case of *Spectral Balance* the difference is much bigger and is due to the normalization used in the feature. In the latest system, this normalization is cut off below a certain energy.

In the case of Attack and Sustain states, the most crucial parameters for the alignment, the difference between the two performances is not huge to the human eye but even the slight difference seen would lead to different system behavior during a live performance, especially for attacks and critical phrases.

Conclusion

In this paper and in the context of a statistical *HMM* score follower developed at IRCAM, we present a new approach for the *observation modeling* which can articulate specific behavior of the musician in a controllable manner. In this respect, the system would be able to grab "stylistic" behaviors of different musicians on the same score and on low-level musical features such as attacks, note sustains and releases.

Using this approach, a novel learning algorithm called *automatic discriminative training* was implemented which conforms to the practical criteria of a score following system. The novelty of this system lies in the fact that this method, unlike classical methods for *HMM* training, is not concerned with modeling the music signal but with correctly choosing the sequence of music events that was performed.

The new system has been tested on contemporary music repertoire such as pieces by Philippe Manoury and Pierre Boulez, and in the case of the former using live musicians. During the tests it has proved to be adaptable to certain style of performance in the case of different musicians performing the same piece and thus, providing a better score following for phrases undergoing changes for different performance styles.

Being rather simple, the system tends to model the margins of different styles of performance to a good extent and moreover, might be a point of departure for further studies in the context of score following and style analysis.

Acknowledgements

We are grateful to Philippe Manoury, Serge Lemouton and Andrew Gerszo without whose valuable comments and presence during test sessions the project could not have advanced. We would also like to thank the anonymous reviewers for their valuable comments on the draft of this paper. We would like to acknowledge Nicolq Orio's ground laying work as the founder of this research project.

References

Bilmes, J. 1998. *A Gentle Tutorial of the EM Algorithm and its Application to Parameter Estimation for Gaussian Mixture and Hidden Markov Models*. ICSI.

Cont, A. 2004. Improvement of observation modeling for score following. Master's thesis, University of Paris 6, IRCAM, Paris.

de Cheveigne, A., and Kawahara, H. 2002. YIN, a fundamental frequency estimator for speech and music. *J. Acoust. Soc. Am.* 111:1917–1930.

Dempster, A.; Laird, N. M.; and Rubin, D. B. 1977. maximum likelihood from incomplete data via the EM algorithm. *Journal of Royal Statistical Society* 39(B):1–38.

Grubb, L., and Dannenberg, R. B. 1997. A Stochastic Method of Tracking a Vocal Performer. In *Proceedings of the ICMC*, 301–308.

Orio, N., and Déchelle, F. 2001. Score Following Using Spectral Analysis and Hidden Markov Models. In *Proceedings of the ICMC*.

Orio, N., and Schwarz, D. 2001. Alignment of Monophonic and Polypophonic Music to a Score. In *Proceedings of the ICMC*.

Orio, N.; Lemouton, S.; Schwarz, D.; and Schnell, N. 2003. Score Following: State of the Art and New Developments. In *Proceedings of the International Conference on New Interfaces for Musical Expression (NIME)*.

Pardo, B., and Birmingham, W. 2002. Improved Score Following for Acoustic Performances. In *Proceedings of the ICMC*.

Puckette, M. 1995. Score Following Using the Sung Voice. In *Proceedings of the ICMC*, 199–200.

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 Music Accompaniment. *Neural Information Processing Systems (NIPS)* (14).

Renals, S.; Morgan, N.; Bourlard, H.; Cohen, M.; and Franco, H. 1993. Connectionist probability estimators in HMM speech recognition. *IEEE Transactions Speech and Audio Processing*.

Reynolds, D. A. 1995. Speaker identification and verification using gaussian mixture speaker models. In *Speech Communication*, volume 17, 91–108.

Vercoe, B., and Puckette, M. 1985. Synthetic Rehearsal: Training the Synthetic Performer. In *Proceedings of the ICMC*, 275–278.