Improvement of Observation Modeling for Score Following

Arshia Cont
Mémoire de stage de DEA ATIAM année 2003-2004
Université Pierre et Marie Curie, PARIS VI

IRCAM, Real-time Applications Group

Under the supervision of
Diemo Schwarz
and
Norbert Schnell

June 30, 2004

Dedicated to my brother, Rama Cont, without whose courage, I would never find myself in Paris.

– Arshia Cont, , June 2004, Paris.

Abstract

Training the score follower, in the context of musical practice, is to adapt its parameters to improve performance for a certain score. To this aim, every parameter used in the system has to have direct physical interpretation or correlation with high-level desired parameters, in order to be trainable and controllable. This criteria has forced us to reconsider the design and approach in one of the main components of the score follower, the *probability observation* block.

In order to this approach, we developed a criticism based on the notion of *heuristics* used in the design of the existing system at the beginning of this project with a look at *empirical-synthetical* sciences which score following research is a member. In his respect, we argue that the heuristics used in the design of the system has been considered in a late stage during the design and suggests an alternative approach in which heuristics will be used as the lowest-level of information modeling and higher-level models used in the system would become an outcome of a series of derivation based on these heuristics modelings.

A novel learning algorithm based on these views called *automatic discriminative training* was implemented which conforms to the practical criteria of a score following. 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. The discrimination knowledge is provided by an alternative algorithm, namely *Yin* developed by de Cheveigné and Kawahara (2002).

Following the design of the training method, further experiments were undertaken to improve the response of system. For this purpose, every feature in the observation process of the system was studied and examined for correlation with high-level states and using the analysis results, modifications on the existing feature as well as a totally new feature were introduced to be used in the system.

During evaluations, the system proved to be more stable and the results are improved compared to the previous system. Moreover, due to our design approach, the shortcomings of the current system have physical interpretations in the terms of current design and can be envisioned for further improvements, which was not the case with the previous system.

Finally, the new concepts presented in this work, opens a new and more flexible view of score following for further research and improvements by arising an urgent need for a database of aligned sound and other research work which would lead the system towards better following.

Résumé en français

L'apprentissage du suivi de partition dans le contexte de la pratique musicale consiste en une adaptation de paramètres permettant d'améliorer le suivi temps-réel d'une exécution d'une partition donnée. Dans cette perspective, chaque paramètre utilisé dans le système doit avoir une interprétation physique et être corrélé directement avec des états haut-niveau désirés, pour que le suivi puisse être entraîné et contrôlé. Ce critère nous a forcé à reconsidérer la conception d'une des composantes principales du suivi de partition: l'*observation des probabilités*.

Dans cette approche, nous avons développé une critique basée sur la notion d'*heuristique* utilisée pour la conception du système au début de ce stage. Pour construire cette critique, nous regardons les principes des sciences *empirique-synthétiques* dont le suivi de partition fait partie. Dans cette perspective, nous prétendons que les heuristiques utilisées ont été considérées dans une étape tardive de la conception et suggérons une approche alternative dans laquelle les heuristiques entrent dans la conception au plus bas-niveau de la modélisation de l'information. Ainsi, les modèles plus haut-niveau deviennent des résultats issus de ces modèles bas-niveau.

Une nouvelle approche d'apprentissage basée sur cette perspective nommée *apprentissage discriminatif automatique* est implémentée. Elle se conforme aux critères pratiques d'un suiveur de partition. La nouveauté est que ce système, malgré les méthodes classiques d'apprentissage des *chaînes de Markov cachées*, ne prend pas en compte la modélisation des signaux musicaux mais la pertinence du choix des séquences des événements musicaux joués. Grâce à l'utilisation d'un processus de *discrimination*, nous tentons de modéliser les marges des catégories au lieu de construire un modèle précis pour chaque catégorie. Ce processus de discrimination utilise un algorithme extérieur du suivi de partition, notamment *Yin* de Alain de Cheveigné et Kawahara (2002). La méthode d'apprentissage proposée est indépendante des paramètres de reconnaissance, ce qui est essentiel pour le développement futur du système.

Après la conception d'une méthode d'apprentissage, des expériences plus approfondies ont été menées pour améliorer la réponse du système. Pour cela, la corrélation entre les paramètres du processus d'observation et les états haut-niveau a été examinée. En utilisant les résultats de cette analyse, nous avons modifié certains paramètres et en avons introduit d'autres dans le système pour une meilleure reconnaissance.

Pendant l'évaluation, le système s'est montré plus stable et les résultats d'alignement ont été améliorés comparativement à l'ancien système. Ces évaluations sont le résultat d'une simulation sur ordinateur et d'expériences dans un studio avec des musiciens. Pendant les expériences en temps-réel, le système

a obtenu des résultats meilleurs pour certaines phrases qui n'avaient jamais été reconnues avec le système précédent. Grâce à notre démarche conceptionnelle, les défauts du système ont des interprétations physiques permettant des améliorations futures, ce qui n'était pas le cas avec l'ancien système.

Finalement, les nouveaux concepts présentés dans ce travail, ouvrent une nouvelle image plus flexible du suivi de partition pour de futures recherches en introduisant des nouvelles demandes comme le besoin urgent d'une base de données de sons alignés. Ils ouvrent de nouvelles pistes de recherche qui doivent conduire le système à un meilleur suivi.

Acknowledgements

I would like to thank my project directors, Diemo Schwarz and Norbert Schnell, who accepted me to their team and have borne with my sometimes extreme excitement and too much self-confidence towards research. Much appreciations to Diemo Schwarz' patience with my impatience towards research and listening to my ideas and loud thoughts, and for being a great supervisor.
Alain de Cheveigné, whom I had the great opportunity to meet during this work, who spent a considerable amount of time explaining *Yin*, giving feedbacks on my approach and widening my horizon of thinking throughout the project.
Philippe Manoury, Serge Lemouton and everyone in the "Suivi Réunion" for their practical feedbacks on score following. Specially to Philippe Manoury as one of the only daring composers always eager to test new developments in score following.
Gerard Assayag, Cyrille Defaye and all the coordinators and professors of DEA ATIAM, first for accepting me to ATIAM, being at IRCAM as one of my dream places since high-school and second for bearing with my strange situation starting from late application and strange requests as the only foreign student in this promotion of ATIAM.
My great ATIAM colleagues, with whom I spent an unforgettable time this year, for their understanding and acceptance of this stranger into their group, for helping me with the language, for all the fun and harsh times during exams, for being such an odd mixture of taste and character, and for sharing their vast experience with me. I will never forget you!
My office-mate and colleague, Nicolas Rasamimanana, whom I could never pronounce his last name correctly, for his extreme patience during work and bearing with my harsh humor, for his flexibility after a while and giving me feedbacks on my jokes after a while! Nicolas, at last we are good friends!
To Thierry Coduys and "La Kitchen" crews in Paris, for accepting me for a part-time position from the beginning of the DEA and keeping me involved with professional music research at all times, without whom I could never manage my life in Paris.
My deepest gratitude to my brother, Rama Cont, and his wife, Rokhsaneh Ghandi, to whom I dedicate this work and without whom I would never had the courage to come to Paris and the accident of getting accepted at ATIAM and my one-year stay would never happen.
To my parents, Gh. Cont and Mansoureh Daneshkazemi and my dearest twin sister, Mandana Cont.
Finally to Marie Vandenbussche for being there, her patience and understanding during this intense project.

Contents

List of Figures

Introduction

This project started with the primary goal of implementing a learning method for *IRCAM*'s score follower. However, it soon changed its direction towards design considerations of the existing system to have a better following which led to a novel training method as well as new component designs in the system. It should be noted that this work is a result of collaborations with the composer Philippe Manoury and his musical assistant Serge Lemouton along with Andrew Gerzso who organized several sessions with soprano Valérie Philippin for testing the score following on live and on *En Echo*.

Before we introduce the contents of each chapter, we would like to emphasize on some terminologies used throughout this report. To this aim, high-level states refer to music symbols modeling the *HMM* system, which are silence and note events (attacks, sustains and rests). Features are essentially audio descriptors marking the first stage of information extraction. Respectively, high-level feature state probabilities correspond to probabilities of high-level states extracted from audio descriptors.

Chapter 1 marks the early studies towards this project, containing detailed studies of other similar systems' implementations followed by a detailed analysis of *IRCAM*'s score follower. In his analysis, the author has undertaken a different view than the articles published in literature on the system in order to emphasize shortcomings and criticize the concepts behind the design of the existing system.

In Chapter 2, a static analysis is described on the system's features introduced in Chapter 1, trying to analyze features' behaviors and their correlations with high-level states. At the same time, a critic on the basis of the existing system is developed which results into modifications presented in other chapters and describes partly the objectives of this project. In this manner, a feature modification and one totally new feature are introduced for the score follower.

A more general analysis on the probability observation component of the system is documented in Chapter 3, which culminates to a redesign and reconsideration of the probability observation based on critics introduced in Chapter 2. The basis of the model and methodology in detail is demonstrated in this chapter.

As a result of the designs and considerations in previous chapters, Chapter 4 introduces a novel training algorithm with subsequent results, called *automatic discriminative training*. The novelty lies in the discrimination section in which we emphasize on modeling feature boundaries instead of trying to model the features themselves.

Chapter 5 demonstrates some evaluations of the new system and comparisons between the previous system in practice. This is continued in Chapter 6 by listing future works which are necessary for further developments of the system

and are in continuation of the current work with a conclusion for this report.

Chapter 1

Previous works and Background

We need not destroy the past. It is gone.
— John Cage

In this first chapter, we aim to give an overview of the previous works which count the early studies undertook for this project. As the first attempt, a history of score following is studied, showing its evolution from the technological and musical side in time and some reflections on the general notion of score following and its outcome in the future. It follows with a more elaborated section on *IRCAM*'s latest score following system which will be the main ground of this work. In studying the *IRCAM*'s score follower, the author develops a subjective scientific view of the topic which would help for understanding the new designs followed in the coming chapters.

1.1 Score Following: A definition in practice

Some remarks on the historical definition of score following will be seen in the next section. However, score following being a medium of interaction between new demands from performers and composers at one side and new scientific technologies on the other side, has changed and evolved in its definition over 20 years. Therefore, a subjective definition can easily loose its account over time. Here we try to define a general definition according to its nature in practice:

> Score following serves as a real-time mapping interface from *Audio abstractions* towards *Music symbols* and from performer(s) live performance to the score in question.

The challenge always lies in how this mapping succeeds and engenders different musical situations in practice such as errors of the musicians and different styles

of interpretation. Over about 20 years of score following research, interestingly, the objectives of this technology have been widened through other disciplines such as Music Information Retrieval among others. However, in our report and research, we rest with the score following used in the context of music performance.

1.2 A brief history of Score Following

Studying the evolution of score following is essential for this work since at all moments throughout this report, we encounter how composers' and musician's expectations along with researchers would help evolve this technology. For this purpose, the author started his work contemplating on the evolution of score following throughout its history leading to the recent notions of score following being a result of almost 20 years of experience and interaction between music and sciences.

It should be noted that while this introduction does not include all the researchers involved in the domain, it tries to lie down most of the main concepts introduced into score following along with introducing their innovators. We have tried to give the least subjective definition on each approach and all the comments on each technology is limited to author's familiarity with the system as well as the literature available. In this manner, Figure 1.1 shows a score following timeline which is gathered and mentioned due to their initiatives, original views and importance in the application and history of score following. We will elaborate on different aspects of each system in the following sections.

In our review, we divide the history of score following in four sections: the early definitions which marks the beginning of score following history, the years of *string matching* and *pitch detection* containing systems using those approaches, statistical approaches and other important systems. After the timeline is over, we contemplate on the important subject of system evaluations and give an overview of the training aspects of each mentioned system, if any.

1.2.1 Early Definitions

The history begins officially in 1984 with Roger Dannenberg's and Barry Vercoe's articles appearing in the International Computer Music Conference (*ICMC*) independently. The two articles mark the first attempts towards real-time score following and real-time accompaniment which would become a major research topic in various research centers and as we will see later, would initiate and mark early attempts for other research topics currently being undertaken in audio research community.

Barry Vercoe's 1984 article titled "The synthetic performer in the context of live performance" defines the objective as follows:

> To understand the dynamics of live ensemble performance well enough to replace any member of the group by a *synthetic* performer (i.e. a computer model) so that the remaining live members can not tell the difference (Vercoe 1984).

While the article describes the system developed in collaboration with Larry Beauregard and for flute, Vercoe pictures the system as having three main elements: LISTEN, PERFORM and LEARN. While he discusses briefly temporal

Authors	Institute	Description	Year
Barry Vercoe	MIT/Ircam	First definition, tempo and pitch considerations, Synthetic performer	1984-1986
Roger Dannenberg	Carnegie Mellon	First definition- String matching algorithm, Pitch oriented with heuristics	1984-1985
Vercoe, Puckette	MIT/Ircam	Training the synthetic performer, string matching added, 'cost'.	1987
Puckett	Ircam	EXPLODE, Pitch oriented	1990
Baird, Belvins, Zahler	Conneticut College	String Matching, phrase matching	1990
Vantomme	McGill University	Temporal Patterns	1995
Dannenberg	CMU	Statistical Modeling	1997
Christopher Raphael	University of Amherst	HMM based score following	1999
Loscos, Cano and Bonada	UPF	HMM based score following	1999
Nicola Orio	Ircam	HMM based score following	2001
Schreck Ensemble	Schreck Ensemble	Neural Network Approach, Pitch based	2001
Pardo, Birminghm	University of Michigan	Pitch based, probabilistic 'cost'	2002
Christopher Raphael	University of Amherst	Bayesian Belief Approach	2001-3

Figure 1.1: Score Following Timeline

modeling of the live performance, his main cue for detection is pitch, which according to the article and the technologies at the time "implies detection at a speed almost impossible for audio methods alone" and thus uses fingering information on the flute. Another interesting contemplation on this early attempt is its author's considerations for *learning* or as he puts it, *learning to improve*. One year later, along with Miller Puckette, he would elaborate more on this topic (Vercoe and Puckette 1985). The learning aspect of Vercoe and Puckette will be studied in a later section.

While Vercoe's approach undertook a "synthetic performer", in his 1984 article, Roger Dannenberg searches for "An On-line algorithm for real-time accompaniment". In his approach, Dannenberg clearly defines his goals as to first detect what the soloist is doing; second, to match the detected input against a score and third, to produce an accompaniment that follows the soloist (Dannenberg 1984). In his approach, he uses dynamic programming to produce the match and consequently concentrates on the second problem above. In his mod-

eling, he considers "*error*" cases which consist of omitted notes as well as extra notes in the sequence. In his matching algorithm, while considering events as string sequences, the *best* match is defined as "the longest common subsequence of the two streams."(Dannenberg 1984) In this manner, Dannenberg's approach can be regarded as a *string matching* technique. It should be mentioned that Dannenberg's approach, too, is dependent on pitch in the soloist event detection. Dannenberg's string matching algorithm ,for which he holds a patent (Dannenberg 1988), is more elaborated in this article by Bloch and Dannenberg (1985).

1.2.2 Years of *string matching* and *pitch detection*

The years that follow the early definition of score following mark several implementations of score following mainly based on *pitch detection* and *string matching* as mentioned above and before the next jump to the statistical approach. Also, we would encounter first attempts to the use of score following in musical composition mainly at *IRCAM*.

Before 1990, Roger Dannenberg and his students would concentrate on improving his string matching algorithm. In (Dannenberg and Mont-Reynaud 1987), they expand previous work by addressing the problem of following solos improved over fixed chord progressions rather than fixed note sequences, leading to new matching algorithms. In order to make the score following more robust, Dannenberg and Mukaino (1988) introduce the idea of using multiple matchers centered at different locations. In this version the system can also deal with trills, glissandi, and grace notes by considering different matching technics for each event. The main low level changes in this version of Dannenberg's score following are the use of multiple matching algorithms at the same time (notion of *matching objects* instead of *matching procedure*) and the use of delayed decisions which prevents accidental matching by not trusting all reports from the matcher and adding a delay of about 100ms to open more decision making opportunities.

The year 1990 marks the appearance of Miller Puckette's *EXPLODE* article (Puckette 1990). In this article, Puckette defines the interface used for score following at *IRCAM* and the score follower itself is more described in (Puckette and Lippe 1992). Puckette's *EXPLODE* marks several pieces written originally with having score following in mind, particularly Philippe Manoury's *Pluton* and Pierre Boulez' *...Explosante-fixe...* . The algorithm consists of pitch recognition along with a pointer to the "current" note as well as a set of pointers to prior notes which have not been matched (Puckette and Lippe 1992). In the 92 article, Puckette and Lippe give an honest report of their result (which is rare in other literatures) noting the weaknesses of the system and when it can not follow perfectly, adding the following comment:

> ... Composers are often forced to make compromises so that their music is followed in such a way that the electronic events in the score are correctly triggered (Puckette and Lippe 1992).

It should be again noted that this version of *IRCAM* score following has no dependency on tempo and makes no predictions about the future behavior of the music to be followed. Rather than use predictions to arrange for the computer and player to act simultaneously (which is Dannenberg's case), the effort was

made to make the delay between the musician's stimulus and the computer's response imperceptibly small (Puckette 1995). This score following had in mind compositions in the score following repertoire at the time of implementation and its assumptions had to be dropped for new composition demands to come, namely Philippe Manoury's *En Echo* for soprano and computer premiered in Summer 1993.

In 1995, Puckette publishes the results of the new approaches to score following as a result of the compromise between technology and new compositional demands (mostly due to Manoury's *En Echo*)(Puckette 1995). While this system, known as *F9* in *Max/MSP*, is still dependent on pitch detection, the instantaneous pitch recognition uses the accelerated constant-Q transform as described in (Brown and Puckette 1992) and (Brown and Puckette 1993). This approach is very similar in its concept to what is being used in the latest *HMM* Score Following's *PSM* (Peak Structure Match) to be elaborated later. In this manner, the best pitch is the instantaneous pitch corresponding to the highest instantaneous power at which a pitch was present (Puckette 1995). For score following, two parallel match signals would be present with different delays and the reliable one would be used as the input to a discrete-event score follower.

In parallel to the above systems, Baird, Blevins and Zahler have revised a new matching algorithm which was based directly on (Dannenberg 1984) and (Vercoe 1984) with the difference that it is based on the concept of segments as opposed to single events. Matching is performed on segments of predefined length; that is, segment sizes are not necessarily based on any musical heuristics or analytic conventions. Comparisons by events and rest positions are performed and stored tentatively until the set of previously heard and unsegmented events match one of four segment types as described in (Baird et al. 1990) and (Baird et al. 1993).

1.2.3 The paradigm of the statistical approach

Contemplating on the nature of score following, we encounter that even with perfect feature observations (pitch, temporal patterns) we always remain in a realm of uncertainty due to various types of errors by the musician or the relative nature of a music performance especially in the temporal aspect. Therefore, it is natural to consider probabilistic approaches towards real-time score following.

The pioneering work in this domain belongs to Grubb and Dannenberg (1997b) for which they hold a patent (Grubb and Dannenberg 1997a). In this approach, the position in the score is represented by a probability density function. Unlike previous systems, it does not require subjective weighting schemes or heuristics and it can use formally derived or empirically estimated probabilities describing the variation of the detected features. In this approach, at any point, the position of the performer is represented stochastically as a continuous density function over score position. The area under this function between two score positions indicates the probability that the performer is actually where in the score. As the performance progresses and subsequent observations are reported, the score position density is updated to yield a probability distribution describing the performer's new location. In this manner, for tracking the performer and to calculate new observation, current score position density and the observation estimations are used to estimate a new score position density.

It is worth to make a comparison between this new paradigm of statistical approach and the previous mainly pitch oriented approaches. In this new approach, if pitch detection is applied to the performance, then this information would provide a *likelihood* that the detector will report that pitch conditioned on the pitch written in the score, despite the previous constraints of score followers highly dependent on the output of the pitch detection.

A simpler statistical approach on the same line as the *string matching* algorithm is reported at (Pardo and Birmingham 2002) and (Pardo and Birmingham 2001) from University of Michigan. Their algorithm is based on the same ideas in (Dannenberg 1984) and (Puckette and Lippe 1992) by defining a *match score* and *skip penalty* with the difference that these "costs" are modeled by some probability distribution instead of mere numbers. In this manner, the system is trained off-line and on a date base of sound (not on the music itself), to obtain a *match score* matrix.

One of the most important works in statistical score following systems has been undertaken by Christopher Raphael in a series of research marking its beginning in 1999. The *IRCAM* Score following system finds its roots in Raphael's 1999 article on "Automatic Segmentation of Acoustic Musical Signals Using Hidden Markov Models" (Raphael 1999b). In this pioneering work and as his first published experiment with real-time score following, he defines musical analogies as Markov models consisting of note models (attack, sustain, duration modeling) and rest models which would be derived from the score and spectral feature observations would be used to calculate state probabilities and decode in real-time. Since there is much similarities between Raphael's system and the system in consideration for this report, we would refer the reader to Section 1.2 and also to Raphael's fully documented article in (Raphael 1999b) and (Raphael 1999a). One last comment on the difference between Raphael's system and that of *IRCAM* is the way the observation probabilities are calculated. In Raphael's system, there is no such distinction between observation probabilities and the *HMM* system and his features are statistics over the data frame.

Similar work on the use and consideration of *HMM* systems for score following have been published at the same time but for limited applications namely in (Loscos et al. 1999a) and (Loscos et al. 1999b).

Raphael would continue on his way and explore more models for automatic music accompaniment which are crucial in the context of our work. In his more recent research papers he reports a new approach by coupling the *HMM* system by a *Bayesian belief network* (Raphael 2001). The basic idea behind this coupling is *HMM*'s famous incapability of handling temporal issues. He uses the output of the *HMM* system as onsets of the *Bayesian network* which would train themselves on a performance sample, thus learning some "interpretational" aspects of the performance and using tempo anticipation in triggering the accompaniment. This issue will be observed in more details later. It should be noted at this point that Raphael's system is pointed more towards accompaniment rather than synchronizing the follower with the performance.

1.2.4 Other approaches

As indicated in the previous subsections, most score following systems described do not consider temporal patterns as main cues for recognition. One of the first attempts to use temporal patterns as the main following cue was done in univer-

sity of McGill by Vantomme (1990). The main input of the system, however, is the onset time of each event in MIDI format and there is few discussion of the liability of this information for the system. Moreover, the implementation is done in LISP which, despite the insistence of its author on unimportance of real-time issues surrounding LISP, should limit processing for certain pieces. Following is based on performer's rhythms which, again despite the insistence of the author on the stylistically unbiased system, should limit its usage specially for contemporary repertoire with non-stationary and complicated rhythmic patterns. However, reports in (Vantomme 1990) indicates that the system is independent of the performer's errors in pitch and robust enough in rhythm recognition, assuming that everything works.

One completely different approach is the *ComParser* score following developed by *Schreck Ensemble* in Netherlands as a free open source software (Schreck Ensemble and Suurmond 2001). Before going into some brief technical details, it should be noted that *ComParser* has been developed to meet its ensemble's musical needs. Therefore, it has certain applications which had not been visioned in the previously mentioned systems such as following *Audio* instead of *symbolic* musical scores. For this reason, the authors of the system refer to their approach as a *sonic approach* as opposed to "*IRCAM*'s Symbolic approach." This system uses the neural network technology for audio recognition with spectrum features as inputs of the network. Although being used in performances of *Schreck Ensemble*, *ComParser* is under development and as of their last report, the network architecture used is a modified *Avalanche* network as originally introduced in (Grossberg 1982) with the addition of time-delays, weights on forward connections and limited lengths and activities of previous activity windows. Like most neural network systems, *ComParser* requires supervised training on audio which can be frustrating, but reports indicate that it is fairly robust. However, use of neural networks technologies introduce problems which would never be encountered in previous "formalized" systems such as premature recognition and over-fitting due to the generalization behavior of the network. On the other hand, recognition of note onset, release and change seems to be not much of a problem and is dependent on training issues.

1.2.5 Evaluation

Speaking about 12 different systems in the previous sections, it is evident to ask about the performance evaluation of systems in general and their advantages and drawbacks among each other. Unfortunately, all the published reports and articles focus on the scientific aspect of the systems and speak less or not at all on evaluation and comparison. This is mostly due to few musical practice in most institutions and non existence of a data base and unified approach for evaluation. However evaluation of score following was a topic of a panel in the *ICMC* 2003 conference but to this date no action has followed that meeting.

One main important issue, not previously addressed, in evaluation of score following is the non-unified definition of score following among its authors. This issue becomes clear by having a closer look at Section 1.2.1, the *early definitions*. From the beginning we see that the term score following is ambiguous between *automatic accompaniment* and *synchronization*. Vercoe and Puckette's approach (or more precisely *IRCAM*'s score following) is dealing with the problem of synchronizing the performer to the score at each instant and on the

other hand, Dannenberg and Raphael's approach is more towards automatic accompaniment. The difference is that in the first approach, we want exact synchronization of the performance with the score and the computer does not impose anything directly on the performer but in the second approach, mostly due to temporal anticipations, future events might occur before the performer's cues and at those moments, the performer must adapt itself to the accompaniment(Dannenberg 2004). This ambiguous usage of the term score following makes the use of literature more difficult and leads to false expectations and comments from each side.

On the other hand, it is at *IRCAM*'s interest to evaluate score following as an *in practice* procedure mostly due to its wider repertoire using score following and musical production environment. In *IRCAM*'s case, the user of the score following is not the developer or researcher and we are always dealing with the tradition of musical practice. Moreover, at *IRCAM* we are dealing with new music repertoire with more demanding and finer score followers than classical music repertoire. Most of the systems described above, besides the *IRCAM* system, are tested and trained using the researchers as musicians and on classical repertoires; thus, not considering their system as an interface dealing with musicians and more over using simpler musical excerpts for score following.

Recently, the Real-time Applications Group at *IRCAM* has brought forward the issue of evaluation in an article published at the *NIME* conference (Orio et al. 2003). In that article, they elaborate the issue by discussing *objective* and *subjective* evaluations, suggesting a framework for evaluation of different existing systems. To conclude, evaluating score following is an essential topic which should be seriously considered for further progress. We would elaborate more on its details in the concluding chapter of this report.

1.2.6 Training in the context of score following

Since one of the main objectives of this project is to obtain an automatic training of *IRCAM*'s score follower, it is worth to look at training in the context of different score following systems observed before.

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

> ... [speaking about the 84 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 and 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 Dannenberg's works before 1997 (or more precisely before the statistical system) there is no report of training, in Puckette's 95 article (*F9* system) 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), Puckette 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.

By moving to the probabilistic or statistical score followers, the concept of training becomes more inherent. In Dannenberg and Grubb's score follower, the probability density functions 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 and Dannenberg 1997b).

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, where there can be many parameters to train and there are traditional ways to train the system, he does not train the *HMM* transitional probabilities. For training his statistics (or features in our system's terminology) he uses the *posterior marginal distribution* $\{p(x_k|\mathbf{y})\}$ to re-estimate his feature probabilities in an iterative manner (Raphael 1999b). In his iterative training he uses *signatures* assigned to each frame for discrimination but it is not clear from the article whether a parsing is applied beforehand to obtain the right behavior or not. In his latest system, incorporating *Bayesian Belief Networks (BNN)*, since the *BNN* 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.

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 and Birmingham 2002).

For *IRCAM*'s score following before this project, training has been done once to obtain the global *PDF*s describing desired behavior and in the context

described in (Grubb and Dannenberg 1997b) as *empiric* and *intuitive* and in (Orio et al. 2003) as *heuristic*. That is series of observations were made on different sound files and musical situation, and using "heuristics" the *PDF* parameters were chosen to obtain desired behavior. It should be noted that in that version of the score following, *PDF*s were fixed exponential functions. There has been no publication to date of an automatic training for this system.

1.3 *IRCAM*'s Score Following

This project is done in the context of *IRCAM*'s score following system as briefly described before. Therefore, as a prerequisite of this work, the current system needs to be studied in depth for further analysis and modification.

In this section we aim to define the architecture which existed upon arrival of the author in the *Real-time Applications Group* at *IRCAM* on March 2004. It should be noted that while the system described hereafter is well documented in (Orio and Schwarz 2001), (Orio and Déchelle 2001) and (Orio et al. 2003), the author's view of the system focuses on different aspects than in the mentioned articles, in order to emphasize more on the learning aspects and shortcomings of the system to be considered during the project and design of a new architecture in Chapter 3.

1.3.1 System Overview

Before anything, it should be noted that while the score following runs in real-time and on audio, it prepares itself for following in advance by loading the score into the system. In general, we can imagine two some-how independent components for the whole system as illustrated in Figure 1.2. In this figure and thereafter, dashed lines refer to information which is processed off-line into the score following.

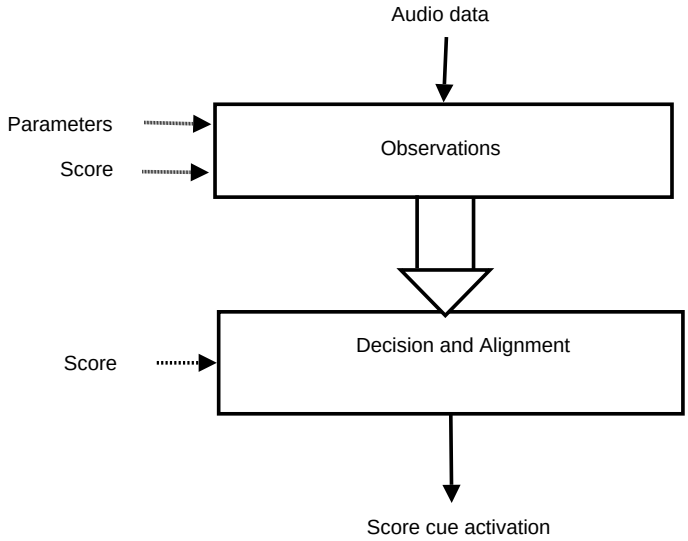

Figure 1.2: General overview of *IRCAM*'s score following

In this view, score following consists of two some-how independent components running in parallel: the *Observation* and *Alignment*. To describe the functionality of each block, it is natural to vision a human listener. In this case, the *observation* is what is being observed on the raw audio by human ears and *alignment* is the high-level information deducted from these observations. *Score* is being pre-processed for both blocks which is basically preparation of an anticipation for observation and preparing the symbolic or high-level targets for the alignment and *parameters* are needed for the observation in order to adapt itself to the situation.

Following this introduction, Figure 1.3 reveals the system with more details on the technologies and terminologies used throughout this report.

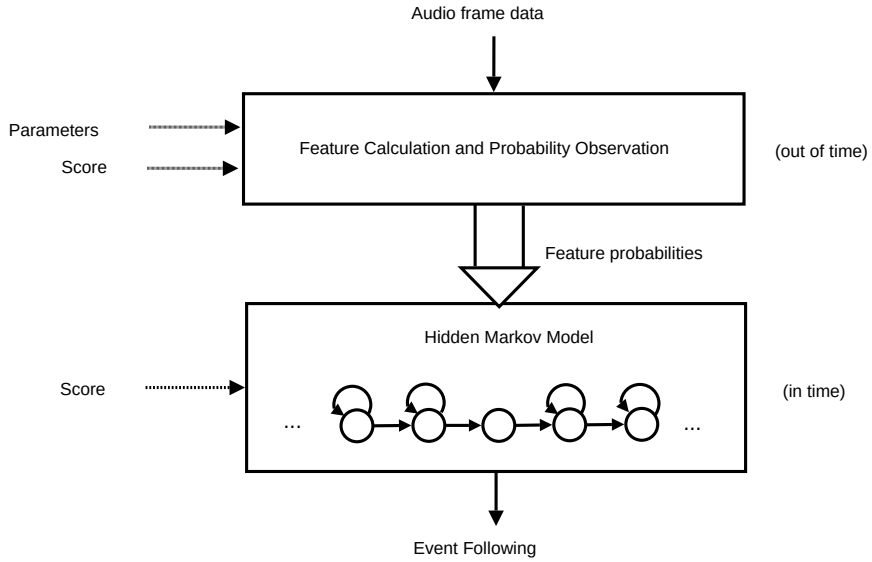

Figure 1.3: More detailed general diagram of *IRCAM*'s score following

Getting more detailed, the observations are *spectral features* calculated on audio frames of about 6ms length and what are actually observed by the alignment section are *feature probabilities* and not the features themselves. In this manner, the observation block in Figure 1.2 consists of both *feature calculation* and *probability mapping* of the calculated features at each instant. As seen in Figure 1.3, Hidden Markov Models (*HMMs*) are used for score modeling and alignment. The score is defined as a Markov model and using feature state probabilities, the system decodes this information into the high-level musical states in the *HMM* which in terms, upon the activation cues marked in the score, triggers outside events.

We can argue that in *observation* we are handling an *acoustic modeling* as in the *alignment* section we are undertaking a *music modeling*.

The score is loaded beforehand into both blocks to prepare the right feature calculation and to construct the *HMM* score model for observation and alignment respectively. Preparation of a *HMM* model from the score is called *score parsing* which is studied in Section 1.3.3. At the same time, and again off-line, parameters are needed to define the probability models for each state features

to be sent to the *HMM* system.

At this point, we are ready to consider each block in Figure 1.3 in more details. From now on in this report, the general score following term will refer to *IRCAM*'s *HMM* based score following.

1.3.2 Features and Observations

Figure 1.4 shows a detailed diagram of the early *Feature Calculation and Probability Observation* section of the score following. This figure shows the real-time process of the mentioned block which undertakes feature calculations and their mapping to high-level state probability observations. Complete lines in the figure refer to number flows as dashed lines refer to vector flows at each instant of score following.

First, we need to define the features being used in the mentioned figure. As is seen, all the features are calculated on the *magnitude spectrum* or *FFT* of the present frame. In this section we go through each feature and through the end, define the probability observation mapping shown in Figure 1.4.

The main ambition in choosing a feature is that each corresponds to one or several high-level states (note sustain, note attack and rest) and that they would be mutually independent among each other. At this point we will not discuss the validity of these features but aim to introduce them for further discussions followed in coming sections. In the score following in question the following main features were in use:

Log of Energy feature (loge)

This feature is simply the Log of the energy contained in the *FFT* frame as shown in Equation 1.1, assuming y to be raw audio signals in a frame. *FFT* in this context signifies the magnitude of the *FFT* of a windowed time frame using a *hamming* window. Main characteristic of this feature is that it will help the system distinguish between note and non-note events.

$$LogE = \log\left(\sum FFT(y)\right) \tag{1.1}$$

Delta Log of Energy feature (Δlog)

The Delta log of energy feature is simply the difference between the current frame's log of energy and the previous one as shown in Equation 1.2. It is hoped that this feature would observe less activity during note sustain and rest and high activity during attacks.

$$Dlog(n) = loge(n) - loge(n-1) \tag{1.2}$$

Peak Structure Match (PSM)

This feature is handling some notion of pitch and at each moment specifies the energy contained at a specified peak structure in the spectrum[1]. More precisely, the pitch is not used as a feature directly but the structure of the peaks in the

[1]This subsection is mainly adopted from Orio and Schwarz (2001).

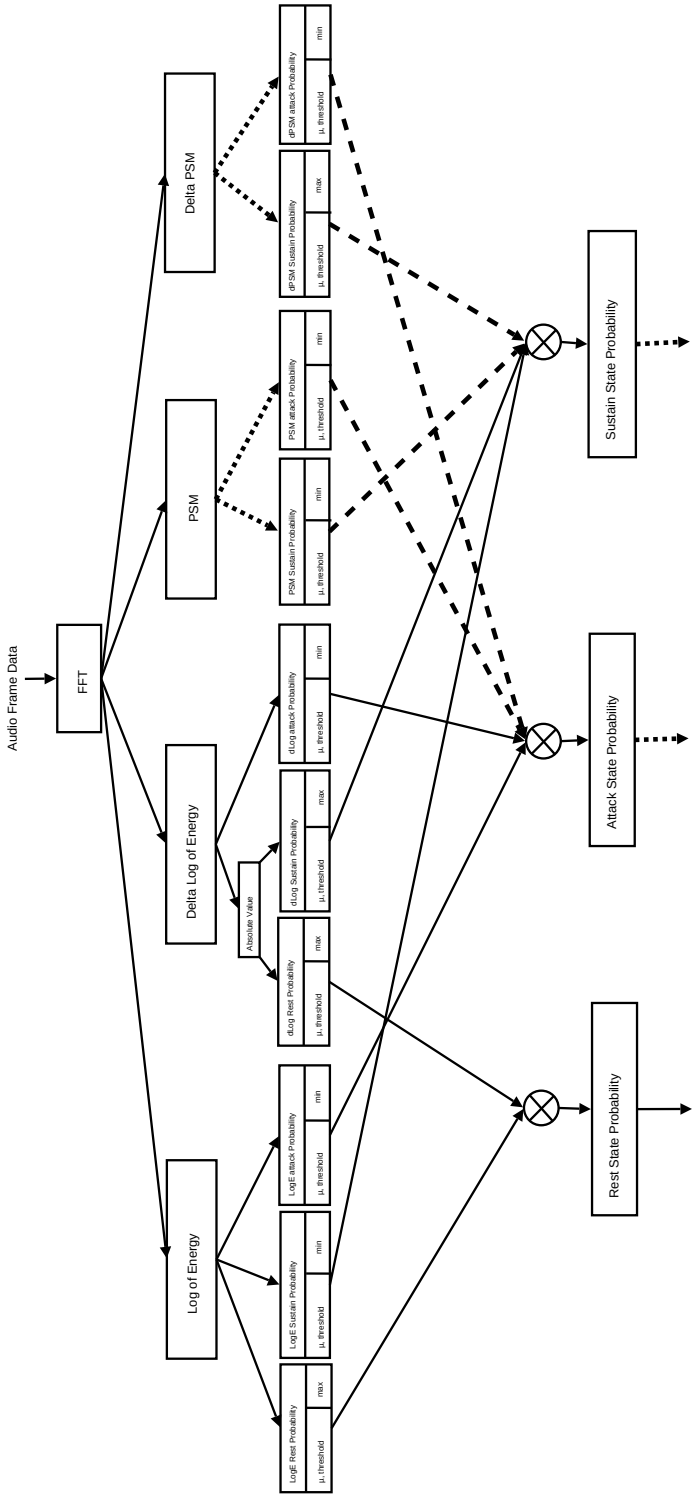

Figure 1.4: Feature and Probability Observation Diagram

spectrum given by the harmonic sinusoidal partials. The advantage is that, this concept can be easily extended to polyphonic signals.

For this purpose, expected peaks are modeled from the pitches in the score. For each note, 8 harmonic peaks are generated. In the latest version, the peaks take the form of trapezoidal spectral bands with an equal amplitude of 1 and no overlap. Figure 1.5 shows a visualization of a one note PSM with different trapezoidal slopes. In the score following, a slope value of 1 is being used with no overlap. Each band has a bandwidth of one half-tone to accommodate for slight tuning differences and vibrato.

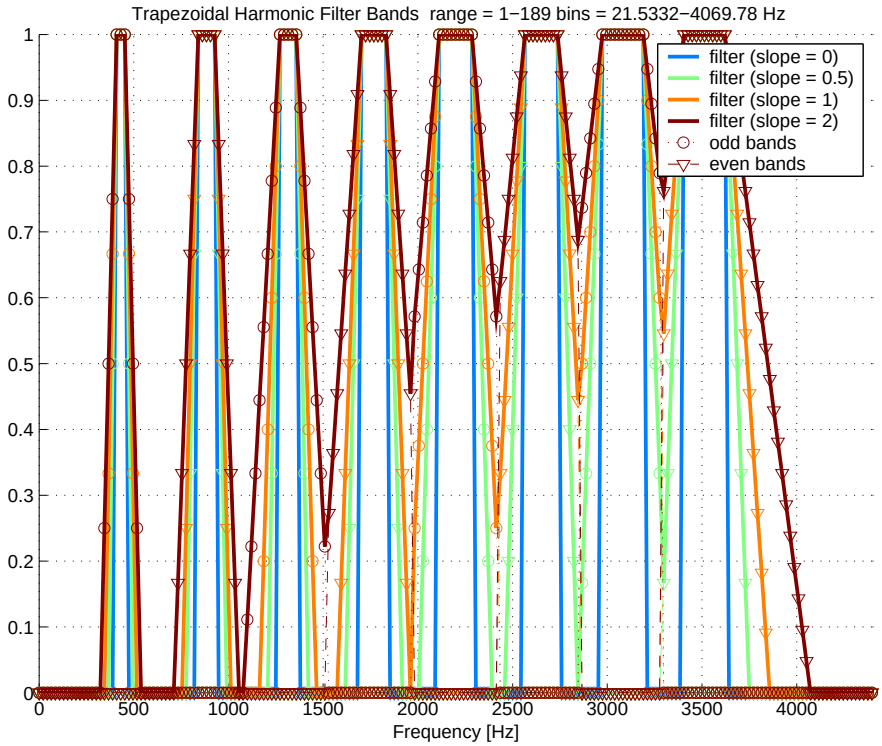

Figure 1.5: PSM Trapezoidal filter banks for one sample note

This generated score spectrum (S) is multiplied by the Fourier magnitude spectrum (P^2) of one frame of audio during performance. Normalization of the result is necessary to prevent a loud, noisy frame from matching all generated bands. Equation 1.3 shows the mathematical analogy of what is described above where m corresponds to the frame number in performance and n is the note event number in the score. Thus, PSM is a vector at each instant carrying information about all possible notes in the score (the reason for having dashed lines in Figure 1.4).

$$PSM(m,n) = \frac{\sum S_i {P_i}^2}{\sum {P_i}^2} \tag{1.3}$$

Delta of Peak Structure Match (ΔPSM)

Like Delta log of energy, the ΔPSM reports local bursts for each PSM and it is there to force and help the recognition of attacks and note changes since we expect to have high rise in PSMs during attacks and also for assuring sustain states since, by heuristics, we expect to observe low activity of ΔPSM during sustain. The equation is essentially similar to that of 1.2.

Probability Observation Mapping

After the features are calculated, their values are being used to compute observation probabilities for high-level states in the *HMM* score model. There are three high-level states at this moment: *Attack, Sustain* and *Rest*. To compute the state probability of the three mentioned states, all or some of the features are used. Before we get to the procedure and practical details of the system as illustrated in Figure 1.4, it is necessary to approach the matter using some mathematics and lie down the assumptions used throughout this process, which would eventually help us in redesigning and training the observation block[2].

A frame of our data (after the *FFT* block), lies in a high dimensional space – $\Re^J$ where J=1024 in our experiment. Thus, we can look at the observation block as a dimension reduction process towards high-level states to make representation and training possible. In this way, we can consider each feature as a vector-valued function, mapping the high dimensional space into a much lower dimensional space. We can think of the features $s(y)$ as containing all relevant information for estimating the desired segmentation, i.e. the alignment l, given no information but y, that is, we assume s is a *sufficient statistic* for l, meaning that $p(y|s(y), l)$ does not depend on l. As a consequence:

$$\begin{aligned} p(y|l) &= p(y, s(y)|l) \\ &= p(y|s(y), l)p(s(y)|l) \\ &= p(y|s(y))p(s(y)|l) \end{aligned} \tag{1.4}$$

Since $p(y|s(y))$ will be constant for each frame we disregard that factor and concentrate on the way we would be connected to the hidden segmentation l. Adding the assumption of conditional independence of each feature (s_d) along, the above equation follows as:

$$\begin{aligned} p(y|l) &\propto p(s(y)|l) \\ &= \prod_{d=1}^{D} p(s_d|l) \end{aligned} \tag{1.5}$$

Which is actually why in Figure 1.4 feature probabilities are being multiplied to obtain the high-level state probabilities. In our case, s(y)= [loge(y), Δlog(y), PSM(y), $\Delta PSM(y)$] with $D = 2 + 2 \cdot N$, N indicating the number of notes in the score, as our feature space.

Knowing this, we are now ready to contemplate on the calculation of each probability density or the details of $p(.)$ functions discussed previously.

[2]The interpretation demonstrated here is partially inspired by (Raphael 1999b) and (Rabiner 1989).

A key design decision in acoustic modeling (given a decision/training criterion such as maximum likelihood) is the choice of functional form for the state output probability density functions. After that we try to adapt these functions to our application by controlling their parameters.

Most *HMM* recognition systems use a *parametric* form of output *PDF*. In this case a particular functional form is chosen for the set of *PDF*s to be estimated. Typical choices include Laplacians, Gaussians and mixtures of these. The parameters of the *PDF* are then estimated so as to optimally model the training data. If we are dealing with a family of models within which the correct model falls, this is an optimal strategy.

For the score following in question the parametric form of *PDFs* is chosen as an exponential function with upper or lower thresholds. Documentations indicate that these functional forms were chosen because they model the *a priori* behavior of the features well enough. In order to better understand the nature of the heuristics used, we demonstrate a simple case of calculating *Rest* probability for the Log of Energy feature: In this case, *heuristics* tell us that the less the energy, the more the probability of being at rest and after some certain value we are sure we would be at rest. Using this reasoning, a lower bounded exponential function is chosen as the probability mapping function in question.

Following the above *heuristics* for every feature, two kinds of exponential probability mapping is chosen to demonstrate an upper threshold (Eq 1.6) and a lower-threshold (Eq 1.7). As is seen in the corresponding equations, for each function the μ and σ parameters need to be adjusted.

$$y = e^{\frac{-(\sigma - x)}{\mu}} \quad \text{for } (\sigma - x) > 0 \qquad y = 1 \quad \text{for } (\sigma - x) \leq 0 \tag{1.6}$$

$$y = e^{\frac{-(x - \sigma)}{\mu}} \quad \text{for } (x - \sigma) > 0 \qquad y = 1 \quad \text{for } (x - \sigma) \leq 0 \tag{1.7}$$

Figure 1.6 demonstrates the general forms of the above probability maps with different parameters.

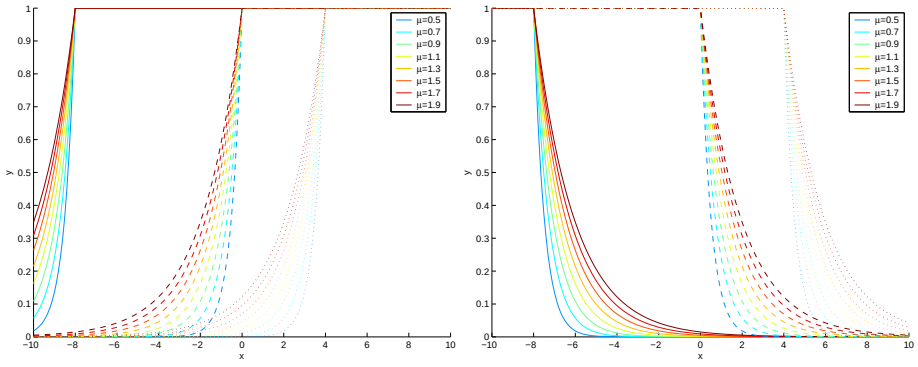

Figure 1.6: Upper (left) and lower (right) threshold exponential CDFs

Looking back into Figure 1.4, the blocks on the second row refer to the mentioned probability mappings which means, having five blocks, there are 10 parameters to adjust.

1.3.3 *HMM* based alignment

As mentioned before, in the *HMM* system the *music model* is being handled from the observed *acoustic model* or it can be viewed as the main bridge from the audio *abstraction* to music *symbolism*. At this point, we describe the process being done in the *HMM* system to align to the right place in the score and the music model used and how it is being mapped in score following.

Hidden Markov Models

Interested readers in *HMM* mathematics and its general model are referred to the historical article of Rabiner (1989). In this section we aim to discuss general issues of *HMM*s which are in direct interest of score following applications.

Hidden Markov Models (*HMM*s) are among the ideal models in literature for sequential event recognition. Score following in its nature is a sequential event recognizer. In a Hidden Markov modeling of audio or music we assume that audio is a *piecewise stationary* process. A *HMM* is a stochastic automation with a stochastic output process attached to each state. Thus we have two concurrent stochastic processes: a Markov process modeling the temporal structure of music; and a set of state output processes modeling the stationary character of the music signal. HMMs are "hidden" because the states of the model, q, are not observed; rather the output of a stochastic process, y, attached to that state is observed. This is described by a probability distribution $p(y|q)$.

In the context of our score following, qs correspond to the high-level state features in the music model which will be described in the coming subsection and probability distributions $p(y|q)$ are the output of our *observation* block as described previously.

For recognition the probability required is $P(Q|Y)$. It is not obvious how to estimate $P(Q|Y)$ directly; however we may reexpress this probability using Bayes' rule:

$$P(Q|X) = \frac{p(y|q)P(Q)}{p(Y)}$$

This separates the probability estimation process into two parts: *acoustic modeling*, in which the data dependent probability $p(Y|Q)/p(Y)$ is estimated; and *music modeling* in which the prior probabilities of sequence models, $P(Q)$, are estimated (Renals et al. 1993). When we use the maximum likelihood criterion, estimation of the acoustic model is reduced to $p(Y|Q)$ as we assume $p(Y)$ to be equal across the model and which is the case in our follower.

Therefore, after the *observation* process described in the previous section, we have all the probabilities required and at this moment, we choose the sequence that has a more likely chance of being the current sequence which in terms, reveals the current place in the score.

Music Model

HMM serves as a bridge from *audio abstraction* to *music symbols*. Moreover, it is the *HMM* which handles time evolution of the score during a performance. For this purpose, a music model should be derived using Markov models.

The model used in score following, considers three high-level states as discussed before: *Attack state, Sustain state* and *Rest state*, which the first two,

obviously are characteristics of a note while the last describes silence and the end of a note. Figure 1.7 shows the Markov model for a single note. It is a *left-right* model which represents time evolution and to this end, the model consists of one attack state, several sustain states modeling the note duration and one rest state which models the end of a note which can jump to the next event (silence or note; note in this example) for the case of *legato* notes.

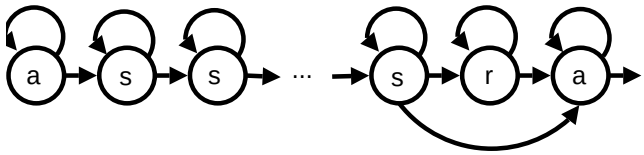

Figure 1.7: Note Markov Model, a=attack s=sustain r=rest

The number of sustain states for a note or the number of rest states for a no-note event as well as the transition probabilities between them models the duration specified in the score. Since this is not the subject of this research, we refer the reader to (Mouillet 2001) which is essentially on the subject of this system's temporal modeling.

Other important parameters in a *HMM* model are the transition probabilities between the states. In the score following, due to the nature of a music performance, only transitions to previous and next states as well as self-transition are allowed which are fixed numbers and assume equal probabilities for each transition (except for time modeling which is the issue in (Mouillet 2001)) and essentially assures a temporal left-right flow in the score, which is natural.

Using this Markov model vocabulary and before each performance, the score is translated into a left-right chain of Markov models. This process is called *score parsing* which is one of the off-line processes discussed before. Figure 1.8 demonstrates a visualization of this process.

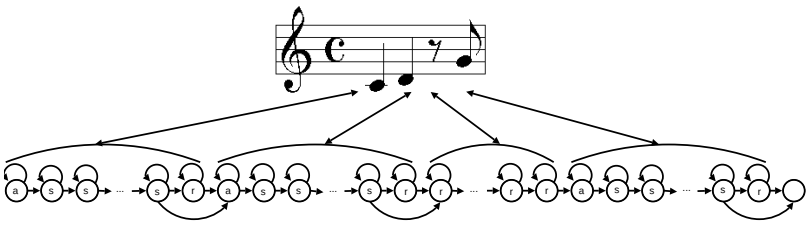

Figure 1.8: *score parsing* visualization

Sequential recognition and Alignment

So far we have discussed the observation probabilities and the music model. While being used in real-time, the score follower is examining the probability that a *sequence* of events would occur. In the classical *HMM* literature an efficient procedure called *Forward-Backward* procedure computes this variable which takes into account both probability of the states before and after a certain state for computation. Clearly, since we are dealing with real-time situations we can have no chance of observing the future states and therefore, we use only

the *Forward Procedure* which takes into account the observation probabilities for all states from the observation block, transition probabilities obtained from the music model and previous sequence probabilities. Classically, this *Forward variable* is referred to as α_t which is basically the probability of the partial observation sequence $O = O_1O_2...O_t$ (until time t) and state S_i at time t, given the *HMM* model λ, as demonstrated in Equation 1.8 (Rabiner 1989).

$$\begin{aligned} \alpha_t(i) &= P(O_1O_2...O_{t-1}, q_t = S_i|\lambda) \\ &= \left[\sum_{i=1}^{N} \alpha_{t-1}(i)a_{ij}\right] b_j(O_t) \end{aligned} \tag{1.8}$$

Note that in Equation 1.8, a_{ij} is the transition probability between states which is obtained from the music model and $b_j(O_t)$ is $p(O_t|q_t = S_j)$ which is a direct outcome of the observation block computed for every state in the *HMM* music model. Also N is the total number of states available in the *HMM* score.

After computing all possible α variables for the audio frame in consideration, we can solve for the individually most likely state q_t, as

$$q_t = argmax_{1\leq i\leq N}[\alpha_t(i)] \tag{1.9}$$

Equation 1.9 is referred to as *decoding* and specifies the most likely high-level state depending on previous matches, new observation probabilities as well as the score and in real-time. At this moment, score following is done!

Chapter 2

Static Analysis

A sound does not view itself as thought, as ought, as needing another sound for its elucidation, as etc.; it has not time for any consideration–it is occupied with the performance of its characteristics: before it has died away it must have made perfectly exact its frequency, its loudness, its length, its overtone structure, the precise morphology of these and of itself.
— John Cage, *Experimental Music: Doctrine*

Experiments with the score following and profound look at the music model in the alignment section reveals that an important aspect of the score following is in the *observation* section. In this manner, we believe that by having a strong *observation* we will always be able to pass to the right state in the music model.

In this chapter, we aim to analyze the behavior of features in the *observation* section and suggest improvements on the features and probability observations to better correspond to the musical events we wish to observe.

We start the chapter by a rather epistemological critic on the basis design of the existing observation block at the beginning of this project, which reveals our methodology in analysis and redesign as well as the approach we believe would lead to a stronger observation. It follows by an analysis of existing features using the mentioned approach and new features would be introduced in Section 2.3.

2.1 A Critic of Pure Heuristics

Heuristics are the main basis of the *probability observation* block of the score following. Although the notion is not quite clear, we try to approach it from a scientific regard and criticize the manner in which it has been used from an epistemological point of view.

In the design of the score following (covered in Section 1.3), the scientific facts

assumed are based on the reproducibility of measurements and the possibility of predicting a result, which makes it possible to validate a theory. In the epistemology literature, this way of integrating scientific data is called *Empirical-Logical* in which the score following lies in the *Empirical-Synthetical* category which is founded on systematic synthesis of the empirical data, interpreted within the framework of the structural level of organization and the emergent functional properties (Castel 1999). This empirical-synthetical method, in epistemological continuity with experimental sciences has its own methodological and scientific validation framework.

In the context of our score following the choice of the probability density functions and features are synthesis of empirical data based on some heuristics assumed dealing with music information retrieval. While the empirical data has been gathered from a database of music sounds, the decision or the *synthesis* of these observations which should culminate to a reproducible and predictable results, based on *heuristics*, is not clear.

2.1.1 Heuristics in features

The heuristics assumed or forced on features are rather simple and straightforward. First of all we need the features to have correlations with the desired high-level states and second, as discussed in Section 1.3.2 on Page 17, features should be mutually independent with atleast no significant correlations among each other so that the multiplication in the last layer of Figure 1.4 on Page 15 would make sense.

In analyzing the features, we would base our exploration on evaluating the two mentioned assumptions. After having the "right" features, we need to evaluate and adapt the probability observations on the calculated features.

2.1.2 Heuristics in Probability Observation modeling

Since we are dealing with statistical modeling in the probability observation calculation and the choice of *PDF*s, we can evaluate the *heuristics* used from a statistical point of view which is a wide topic in statistical modeling and estimation of observed data (Pollard 2001).

The dogma used in estimating feature observation probability in the score following is that each feature is modeled with an exponential probability distribution with μ and σ parameters to control as demonstrated in Section 1.3.2. Two questions arise while studying the behavior of this choice as a probability mapping:

- Whether the choice of exponential probability provides enough flexibility for high-level state observation and how we have derived this form of representation?

- How would the parameters μ and σ correspond to physical phenomena for adapting the functionality to a certain performance?

As mentioned before, the choice of the exponential probability mapping is due to logarithmic descending and ascending probabilities of feature behavior followed from heuristics. While the author of this article does not criticize

the use of heuristics in the case of an empirical-synthetical implementation, he criticizes the stage these heuristics are used for the mentioned modeling.

For the score follower in question, heuristics impose the probability model and the parameters are to be optimized by looking at observations. Without hesitation, this methodology produces a loop of evaluations on the model since we use observations to fit a model which has apparently been derived from considering observations along with heuristics. Moreover, the heuristics in defining this model have entered in a very high-level stage. The author believes the natural way of dealing with such empirical-synthetical situation is to impose heuristics at the lowest-level of information modeling and construct the higher-level models on this ground basis information. In this manner, the probability model will not be directly a result of any heuristics but a series of derivations which in their roots are based on some heuristics.

This argument will become clear when trying to control the μ and σ parameters as σ clearly signifies the threshold for a certain detection but μ does not find any clear physical interpretation for training; which simply means that the heuristics in constructing these models have not been well-defined. A suggested training method for exponential *PDFs* will be covered in Section 4.2.

We have based our redesign of the probability observation block on the above methodology which conforms to a empirical-synthetical approach as well as the special case of score following.

2.2 Feature Analysis

Training and adapting parameters would make sense when features correspond to the desired high-level states. If we adapt all parameters and the features do not behave as desired, we can not have an acceptable score following. Therefore, eventhough the main objective of this research project was training the score following, we examined the features at the same time and what is being presented are contemplations of the mentioned work.

For this purpose, we examine two features which seemed to be most problematic for certain high-level states[1]: the Delta Log of Energy and the ΔPSM.

An ideal feature should observe atleast three behaviors:

- High correlation with the appropriate high-level state
- Mutually independent from other features
- *Stability* in the sense that during the appropriate high-level state event, we should not observe steep changes.

Using the above criteria we examine the two mentioned features. The experimental results shown are feature observations on three pieces that use score following extensively: Phillipe Manoury's *En Echo* for soprano and electronics, Pierre Boulez' *. . . Explosante-Fixe . . .* and Philippe Manoury's *Pluton* for piano and electronics.

One problem of the score following even with trained probability mapping is that we observe early detections such as jumps to future states at unwanted

[1] Experience and the ground basis of score following show that the two other features (Log of Energy and PSM) are quite stable and are good candidates for forcing correct passes.

times such as during a note sustain. This is because the probability of another note's attack state becomes suddenly higher than the sustain probability at an inappropriate time. Our experience shows that this inconsistency is mainly produced because of:

1. Inconsistency of the two delta features with the appropriate feature, and
2. Instability of the two delta features leading to sudden jumps when there should not be any.

Here, we demonstrate these inconsistencies of the two delta features:

2.2.1 ΔLog of Energy

From its original conception, we expect that the Δlog energy would observe high activity for *Attack* states and low activity during *Sustains* and *Rests*.

Figure 2.1 shows the Δlog energy features observed for measure 39 of "Riviere", the first movement of *En Echo* for soprano and electronics as one of short phrase, summarizing most of the score following problems: repeated notes in the beginning and sudden jumps during the following. The red features correspond to the beginning of a note, extracted with the aid of *YIN* (de Cheveigné and Kawahara 2002).

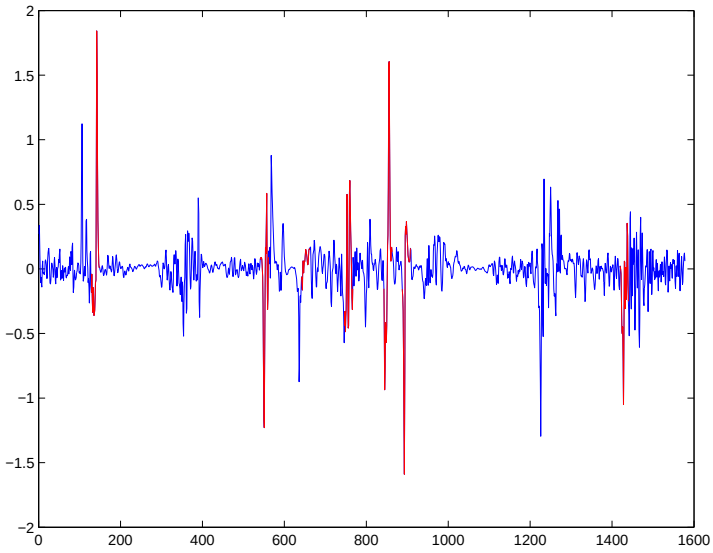

Figure 2.1: Discriminated Δlog energy feature for measure 39 of RIVIERE

From the figure we can see that large feature changes occurs when there is no attack (in this case sustain). The reader might argue that these changes are not as large as the ones observed during attacks but we remind that first of all, due to the nature of the feature, the probability model of this feature would be very steep leading to big change in probability by observing a small burst, and secondly, an increase in Δlog energy implies an increase in attack state

probability and decrease in sustain and rest states at the same time. This experiment among with other similar results brings out immediately the question of stability and right correlation with the high-level states.

In order to make things more clear about the above argument, Figure 2.2 shows a histogram of the Δlog features observed on note sustains and only for values over 0.4 and on the entire movement of "Riviere." As mentioned before we expect to observe none or very few high activation of this feature during sustain states which is clearly not the case.

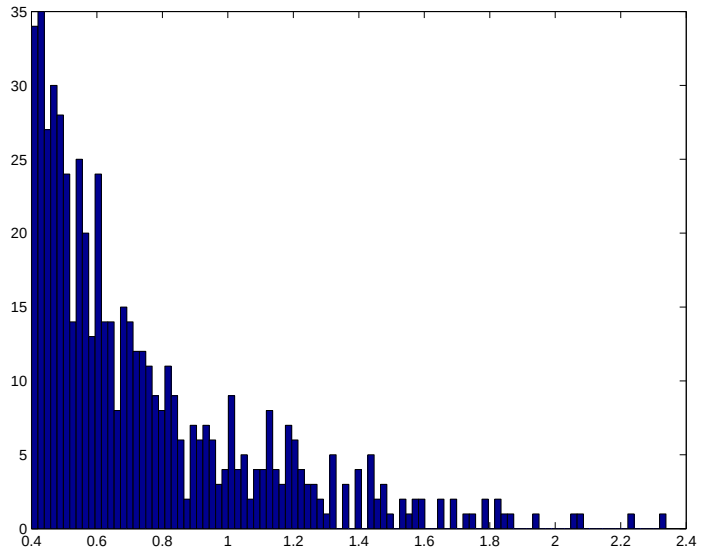

Figure 2.2: Δlog energy feature histogram during note sustains, for $\Delta\log \geq 0.4$

2.2.2 ΔPSM

Like Δlog energy, the ΔPSM feature helps in differentiating between sudden bursts and mainly signifies note transitions. One quantitative difference between this feature and Δlog energy is that for each analysis frame it is a vector rather than a number, representing delta operation applied to each *PSM* feature of each note in the score.

Figure 2.3 on Page 28 shows two instances of ΔPSM feature again on measure 39 of "Riviere." Here again, the red color represents the beginning of a note and we expect to observe high activity only during this time and stable low activity at other times.

While the same arguments for Δlog energy hold here, we see that they are more exaggerated to the extent that we can question the foundation of this feature for having the right correspondence with the desired high-level states. Being very unstable, this feature can not be a representative of attack states as well as sustain states.

Moreover, we can argue on the nature of this feature, being the delta of the PSM representing the pitch structure, that a simple Δ function and on the same pitch, can in no way represent changes in high-level states for several reasons:

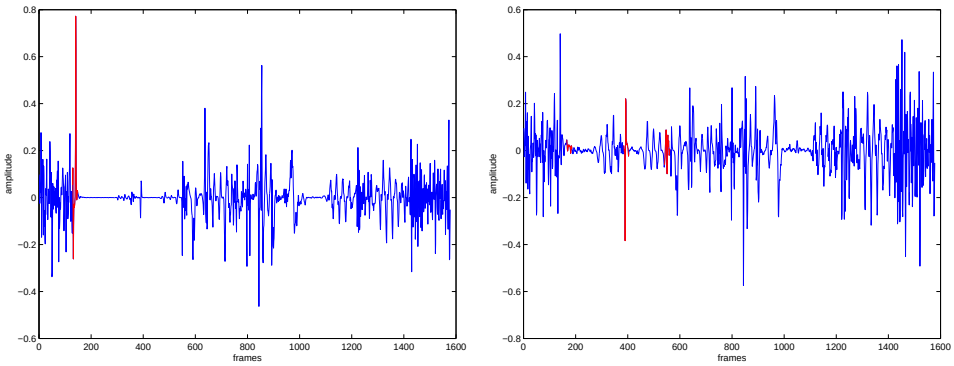

Figure 2.3: ΔPSM for note 1 and 2 on measure 39 of RIVIERE

- At each instant during the performance, the presence of partials of other notes in a harmonic spectrum of a different note would cause changes in PSM and subsequently bursts in ΔPSM.

- On the other hand, in computing a notion of difference in the case of PSMs, it is natural to use some notion of distance between all notes whether here the difference is being calculated on the PSM in question itself without any consideration for PSM values of other notes.

2.3 New Feature Considerations

Following our observations and contemplations on two problematic features, we attempted to refine the features as well as introduce features that would better meet the criteria described in Section 2.2. Here we represent a refinement on the Δlog feature after series of experiments and also a completely new feature which so far best meets the criteria mentioned.

2.3.1 Moving Average ΔLog of Energy (mdlog)

As one main problem of the Δlog energy feature is its instability, we try to make the feature smoother by considering not only the previous frame information for calculating the delta but a frame window. As a natural choice, we used a moving average process which is defined as in Equation 2.1 which can also be regarded as a filtering on the feature.

$$mdlog[n] = \frac{1}{M} \sum_{k=1}^{M-1} dlog[n-k] \tag{2.1}$$

Figure 2.4 shows the moving average Δlog feature that corresponds to the same sample of Figure 2.1 on Page 26. Comparing the figures would reveal improved stability of the new feature still despite the presence of unwanted peaks.

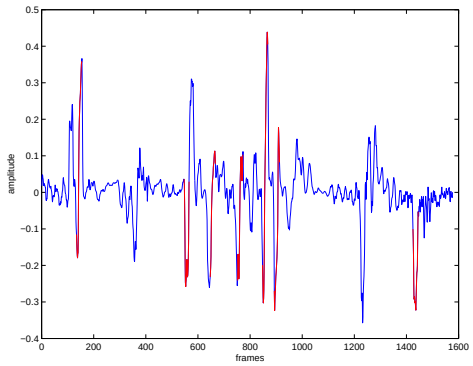

Figure 2.4: Moving Average Δlog energy for measure 39 of Riviere

One drawback of the moving average filtering is the introduction of delays. In this manner, the discrimination undertaken for Figure 2.4 has a delay of about 5 frames compared to the discrimination done for Figure 2.1.

Although the moving average approach has improved the earlier feature, still the same arguments for Δlog energy holds but in a slighter manner. Therefore, some efforts were taken in looking for new features that would resolve the problems observed with Δ features.

2.3.2 Spectral Activity Feature

As one of the last attempts in this research project, we tried to implement a feature consistent with the criteria defined in 2.2 to have a more stable, correct and smooth score following. As a result, we introduce a new feature called *Spectral Activity* feature as described in Equation 2.2 where it is applied to each frame's *FFT* output and J signifies the number of *FFT* points.

$$SpectralActivity(y) = \frac{(\sum_{j=1}^{\lfloor J/3 \rfloor} y^2(j) - 2\sum_{\lfloor J/3 \rfloor+1}^{\lfloor 2J/3 \rfloor} y^2(j) + \sum_{\lfloor 2J/3 \rfloor+1}^{J-1} y^2(j))}{\sum_{j=0}^{J-1} y^2(j)} \tag{2.2}$$

From a mathematical point of view, this feature provide us with a measure of the spectrum's curvature at that frame and the physical interpretation is a measurement of spectral *burstiness* of the signal. A closer look at the equation reveals that we are computing different levels of energy in three spectral bands.

Figure 2.5 shows the spectral activity feature again on the same excerpt used before (measure 39 of "Riviere").

The feature's behavior is much more stable than the delta features discussed before and it directly distinguishes attacks, sustains and rests. More interestingly, as is demonstrated in Figure 2.5 it has been able to distinguish the repeated notes (a classic problem of score following since its conception) that *YIN* algorithm could not recognize. Again in this figure, the red color corresponds to the beginning of the note.

To provide more comparison, Figure 2.6 demonstrates the histogram of the spectral activity features during *note sustains*. Clearly, the values of this feature

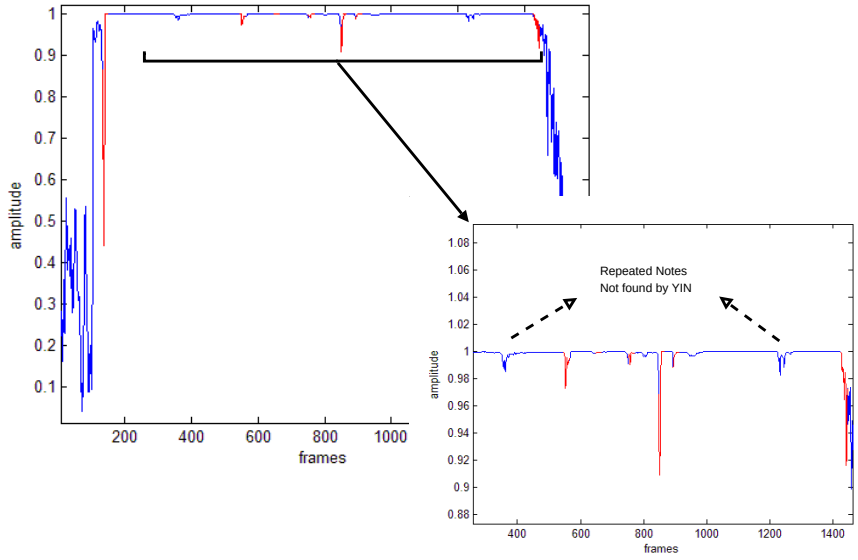

Figure 2.5: The new Spectral Activity feature on measure 39 of Riviere

are highly concentrated near 1 which demonstrates the desired stability for score following as opposed to our previous observations in Figure 2.2.

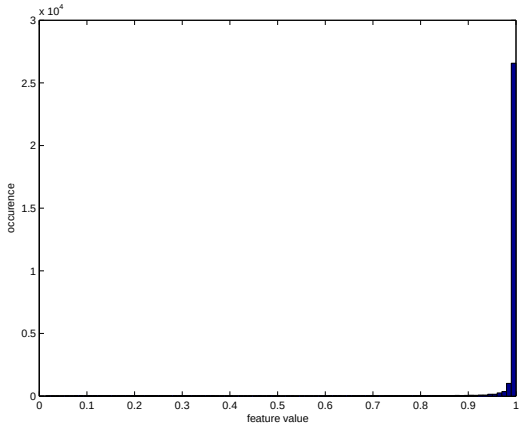

Figure 2.6: Spectral Activity feature histogram during note sustains on the entire Riviere

Moreover, since the spectral activity feature is strong enough to distinguish between the three high-level states, we were able to safely remove the two Δ features from score following, since we doubted their direct correlation with high-level states as well as enough stability, without any need for more features. Therefore, in the version of score following using the *Spectral Activity* feature, the total number of features are reduced to three: Log of Energy, PSM and Spectral Activity features.

Chapter 3

Probability Observation Design

One of Mies van der Rohe's pupils, a girl, came to him and said, "I have difficulty studying with you because you don't leave any room for self-expressions." He asked her whether she had a pen with her. She did. He said, "Sign your name," She did. He said, "That's what I call self-expression."
— John Cage, *Indeterminacy*

Having acceptable features, we need to have a *probability observation* that would be well-defined on both the features and the high-level states. Having a well-defined architecture would allow us to better plan a training leading to desired behavior as will be discussed in Chapter 4. In this chapter, we first go through the basis of our modeling, describing how our model's basis beliefs diverges from that of the previous system. It follows by an overview of the probability functions used in the design and the notion of *discrimination* used widely in this report. After this introduction, a statistical analysis will study and model each feature based on the described basis. Finally, the new designs for *Probability Observation* block of score following are demonstrated along with some remarks.

3.1 Model basis

The model basis suggested here is what follows directly from critical thoughts in Section 2.1.2 on Page 24. In those lines, we envisioned heuristics to enter the model at a low-level stage in order to have more control over parameters and behavior of the system.

In the previous model, all probability mappings are modeled by exponential

functions as a result of experience and heuristics and ideally, its parameters should be controlled to fit to an observed model. But as discussed in Section 2.1.2, the insufficient definition of this probability mapping would lead to training algorithms which are mathematically and physically hard to interpret with suboptimal results in the end as will be demonstrated in Chapter 4.

In our methodology, we force no *a priori* probability mapping. We believe that in its nature, a probability mapping is *a posteriori* and should be derived from low-level observations. In this way, our *a priori* data would be several observations on recorded music and we would fit some pre-defined probability models, obtained from statistical analysis of several observations, and construct the probability mapping using each feature's *heuristics* and in the end of the process.

In this manner, the new model requires a statistical analysis of feature observations by *synthesizing* the *empiric* data into probability models which we assume would predict and reproduce future events. This way, the probability mapping will be constructed as a *final outcome* rather than an *a priori* model.

Before we get to the statistical analysis and modeling, we would give a summary of the probability preliminaries used as well as the notion of discrimination which is used widely in this report.

3.2 Probability preliminaries

In this section, we give an overview of the probability concepts and functions in use in the new probability observation block. In general, after statistical analysis of features, we would model each high-level feature observation with a *gaussian* or *normal* distribution. Equation 3.1 demonstrates the *gaussian probability density function (PDF)* which is demonstrated in Figure 3.1.

$$P = f(x|\mu, \sigma) = \frac{1}{\sigma\sqrt{2\pi}} e^{\frac{-(x-\mu)^2}{2\sigma^2}} \tag{3.1}$$

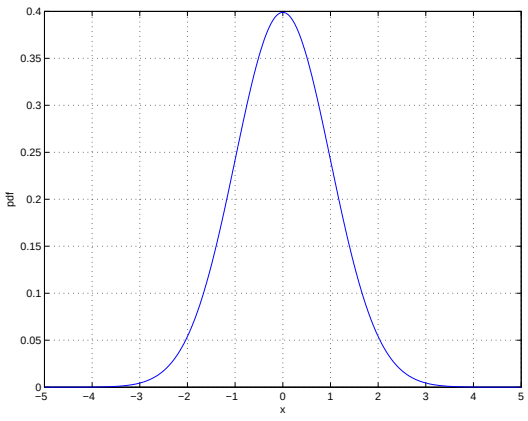

Figure 3.1: A Gaussian PDF with $\mu = 0$ and $\sigma = 1$

After modeling the feature states, we model the probability mapping for each high-level feature states. For this stage we make use of *Cumulative Distribution*

Function(CDF) and its inverse associated with each modeled *gaussian PDF*. Equation 3.2 and 3.3 demonstrate the calculation of *CDF* and *inverse CDF* respectively from a *PDF*. Note that the *inverse CDF* is simply $1-F(x)$. Figure 3.2 visualizes both functions for the same *PDF* in Figure 3.1.

$$F(x) = \int_{-\infty}^{x} f(t)dt \tag{3.2}$$

$$F_{inv}(x) = \int_{x}^{+\infty} f(t)dt \tag{3.3}$$

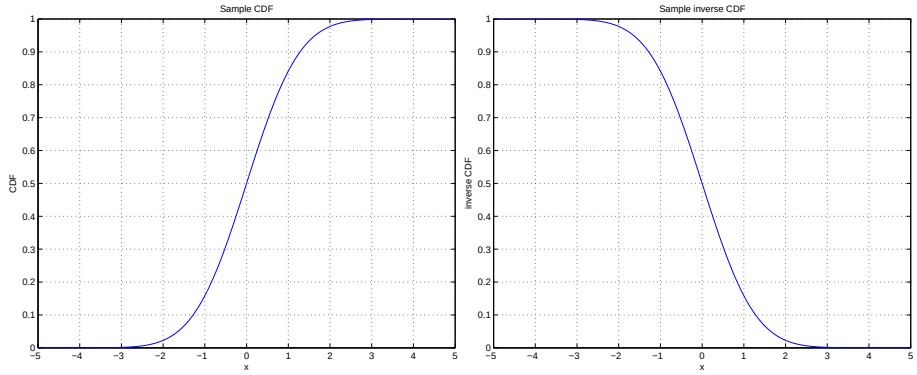

Figure 3.2: CDF and inverse CDF samples with $\mu = 0$ and $\sigma = 1$

For choosing the right function for the *probability mapping* we make use of the *heuristics* associated with each high-level feature states. In this manner, a *CDF* is chosen for state features which observe high probability for higher values (case of all sustain features) and *inverse CDF* for state features observing high probability for lower values (case of rest features) and the *gaussian PDF* itself when the value should be concentrated around some observations.

To conclude this section, we note that the notion of *PDF* has been used in the previous system for probability mapping, but mathematically, the exponential functions used are not *PDFs* but *CDFs*. Moreover, use of *gaussians* rather than exponential functions leaves us with more physical interpretation of parameters and more precision during training.

3.3 Introducing Discrimination

When trying to model each high-level state in the features, we note that while distinguishing note events and rest events are not complicated, distinguishing between higher-level note events (that is attack and sustain states) is not that straightforward. For this reason, we introduce a process of *discrimination* on note states, distinguishing between attacks and sustains whenever there is a note event. We will elaborate more on the foundations of this notion in Section 4.3.

For this project, *Yin* f_0 estimator (de Cheveigné and Kawahara 2002) is used for discriminating high-level states. At each frame, *Yin* provides us with an estimation of the aperiodicity of the time-domain signal as well as an estimation

of f_0. Using this information, we can easily distinguish between rest and note events and further more, by putting more constraints on the signal, the attacks and sustain frames. The imposed criteria for discriminating attack is the beginning of f_0 events which exist in the score with an imposed minimum duration. During our experiments, we have concluded that with our analysis parameters, taking about 15 frames at the beginning of each mentioned f_0 would give an acceptable estimation of attack feature values.

We should note that we only need a *not-so-precise* approximation of features out of discrimination since we are working with statistics. That is, eventhough *Yin* seems to be working very well in our case, introduction of errors in discrimination would not harm our modeling. Finally, note that since *Yin* works in time-domain, we would adapt its parameters so that there would be a one-to-one correspondence between our feature frames and that of *Yin*.

3.4 Statistical Analysis and Modeling

As stated, we tend to observe the features for as many musical situations as possible (*empirical*) and construct a probabilistic model upon them to reproduce desired behavior and predict future observations (*synthetical*).

The ideal approach to this empirical-synthetical method is a huge data base of *aligned* music files, specially the ones which use score following in performance. While there is a huge need for such a database, it does not exist up to the time of writing this report. Therefore we have used music files which exist in the score following database, that is music which uses score following in performance. While the results are promising and are currently in practice, having such database is necessary for further progress in score following as will be discussed in Chapter 6.

In this regard, we aim to observe statistical behavior of features in high-level states. That is we observe how each feature reacts during rests, attacks and sustains and decide upon modeling each high-level feature probability modeling. After this modeling, using each feature's *heuristics* (as demonstrated in Section 3.2) we define the probability mapping associated. For each case, we demonstrate one or two examples as a report of the entire work.

3.4.1 Log of Energy Feature

Since log of energy feature is directly correlated to the distinction between note event and rest event, it can be observed directly without need for further discrimination of high-level states in the feature. However, for *Attack* observation we would always need discrimination as described in the previous section.

Figure 3.3 on Page 35 shows three instances at which this feature has been calculated for an entire piece with the histograms produced. We can see that, for each histogram presented there are two places where population is centered. The lower value population corresponds to *rests* in the piece (silence and background noise) and the higher value population corresponds to *note events*.

Moreover, by synthesizing this concept and looking further at the shape and population we can assume to model each high-level event in this case by a *Gaussian*. By this assumption, we try to fit one gaussian to each population

representing note and rest events as is demonstrated for one histogram in Figure 3.3.

For modeling the high-level state probability mapping using the derived *gaussians*, we use the same heuristics used in the previous system but in a more mathematical rigor: For rests, an *inverse CDF* would fit the heuristics that for less energy, the rest probability is here. Conversely, a *CDF* would model both sustains and attacks.

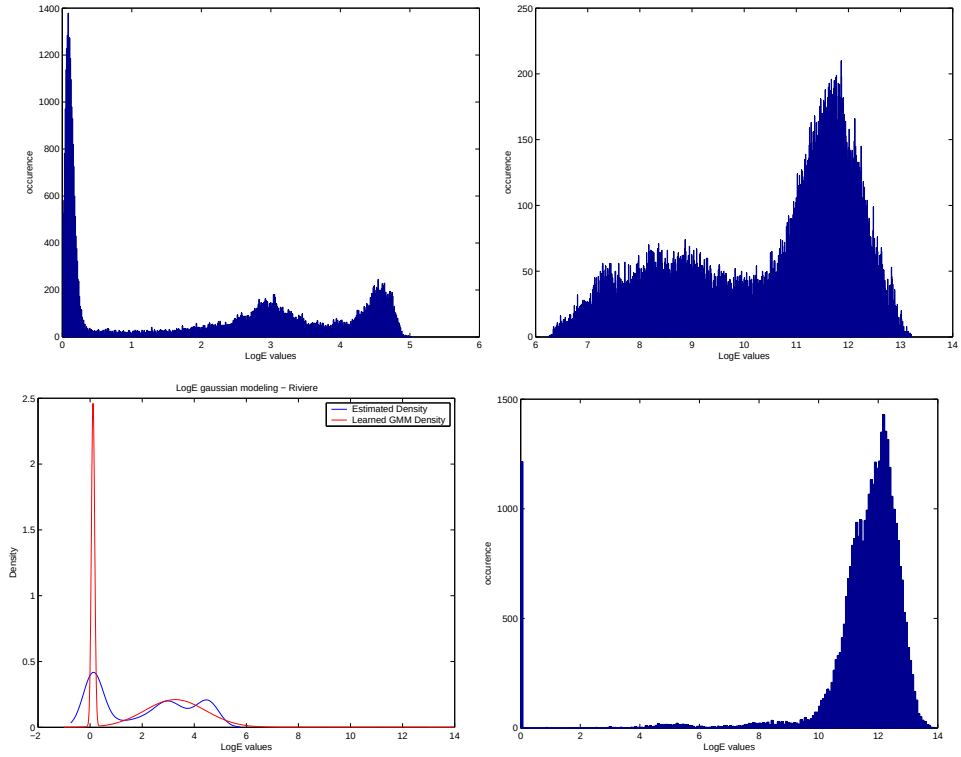

Figure 3.3: Histograms of LogE for "Riviere" (top-left), "...Explosante Fix ..."(top-right) and "Pluton" (low-right) with estimated gaussians for "Riviere" (low-right)

3.4.2 Spectral Balance Feature

Looking at the Spectral Balance feature's histogram, we can still distinguish the rests and notes but not higher level note states as is the main job of spectral balance feature.

Figure 3.4 on Page 36 shows the histogram of spectral balance done on "Riviere" as well as results of discrimination for all three high-level states. As for Log Energy feature, we can model each event using a *gaussian*.

After modeling each high-level feature probability with a gaussian, as before, we model the final probability mapping uing features' *heuristics*. In this respect, for *rests* an *inverse CDF* would model the right probability as the least activity

suggests higher probability. Conversely, a *sustain* is modeled with a *CDF*. In the case of *attacks*, we need the probability to raise as activity increases but low down as we are in sustain. Therefore, for *attack* observation, we use the same *gaussian PDF* that models the statistics.

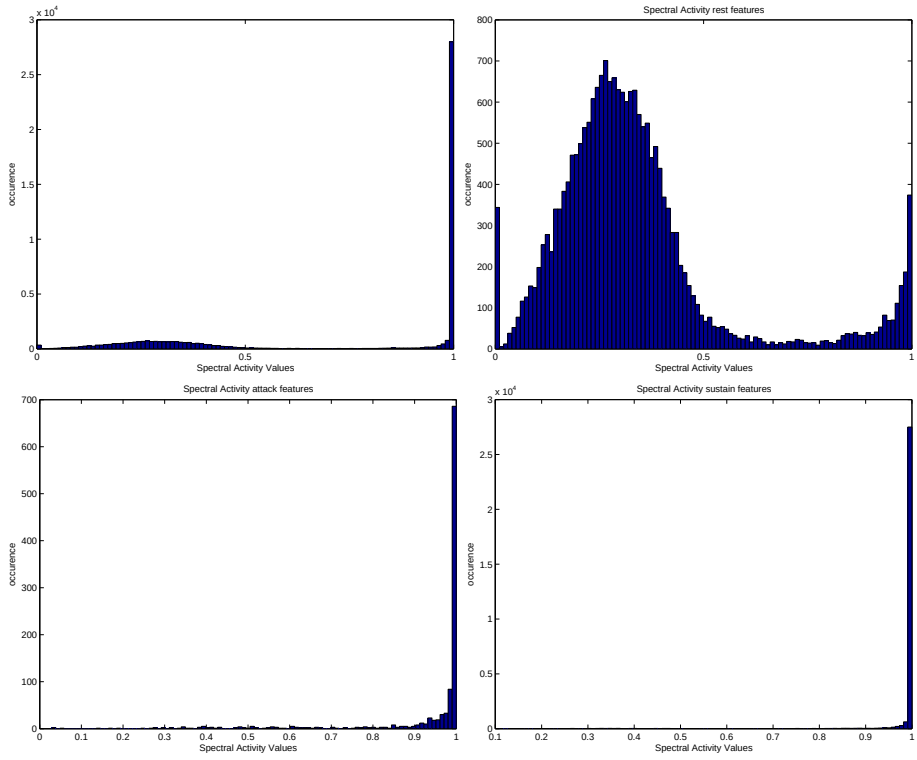

Figure 3.4: Histograms of Spectral Balance (top-left) and Discriminated Spectral Balance for high-level states (in "Riviere")

3.4.3 PSM feature

The same procedure followed above applies to *PSM* features with the difference that, as mentioned before, *PSM* features come in vectors in which each value presents the feature value for one note in the score and also, as its nature suggests, *PSM* will not be used for rest probability.

Figure 3.5 (left) shows the *PSM* feature calculated on the entire "Riviere" where the feature values are presented for all different notes in the score. Note that a pitch match in *PSM* means a value of very close to 1 and we see clearly that the population around 1 is not very high since this figure is generated without discrimination. In order to further discriminate this feature we use *Yin*'s f_0 output plus a filtering method which would allow us to extract the right notes at the right place. For this purpose, as shown in Figure 3.6, for every note present in the score we construct a bandwidth and we extract the frames' information that lie within each interval which in terms corresponds to feature values for a certain note in the score. Figure 3.5 (right) shows the result

of this discrimination on the same feature values.

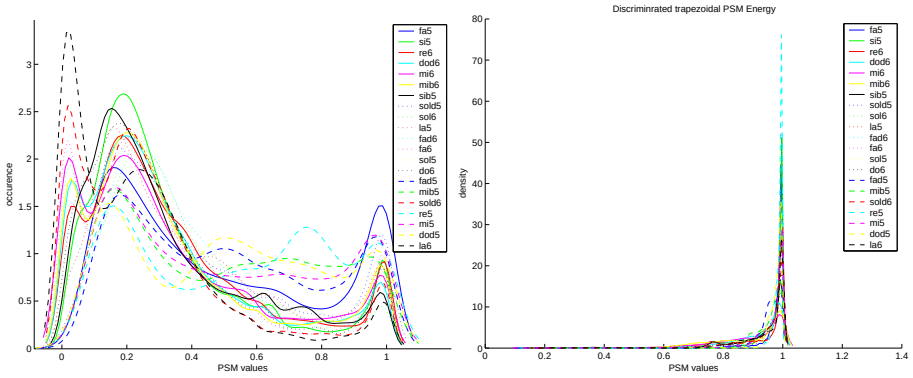

Figure 3.5: Histogram of *PSM* feature non-discriminated (left) and discriminated (right) for all notes in "Riviere"

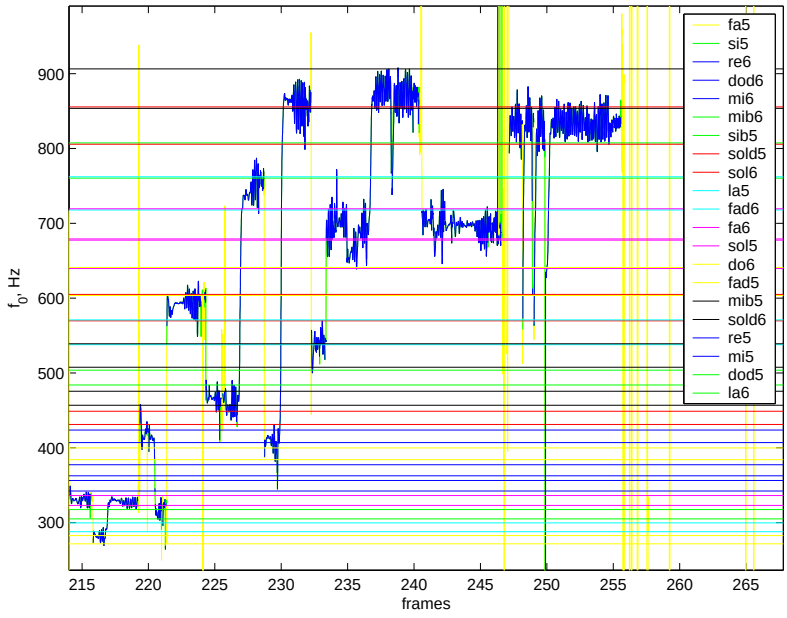

Figure 3.6: *PSM* discrimination using *Yin*

This discriminated data will be used to model sustains and attacks using two gaussians. In this respect, the probability mapping for both features would be *CDFs*.

3.5 Design Summary and Remarks

Integrating the suggested models for probability observation and mapping, the new *probability observation* block would become as demonstrated in Figures 3.7

on Page 39 and 3.8 on Page 40 for different feature suggestions. Features used in each diagram correspond to our discussions in Section 2.3. In Figure 3.7, the *Spectal Balance* feature has replaced both *Delta* features and in Figure 3.8, *Moving Average* Δlog feature has replaced Δlog feature.

In the new system, each probability mapping block (2nd layer) is loaded before performance with its appropriate trained *gaussian model*. We see that after this modeling and above discussions, training gains a physical meaning and becomes integrated into the system. Training will be discussed in details in Chapter 4.

The two proposed systems are the final outcome of this project and the right choice between the two depends on *Robustness* results of the two systems for which experiments are underway to use in concerts. So far, the system with *Spectral Balance* feature has proven be more robust and leading to better results than others.

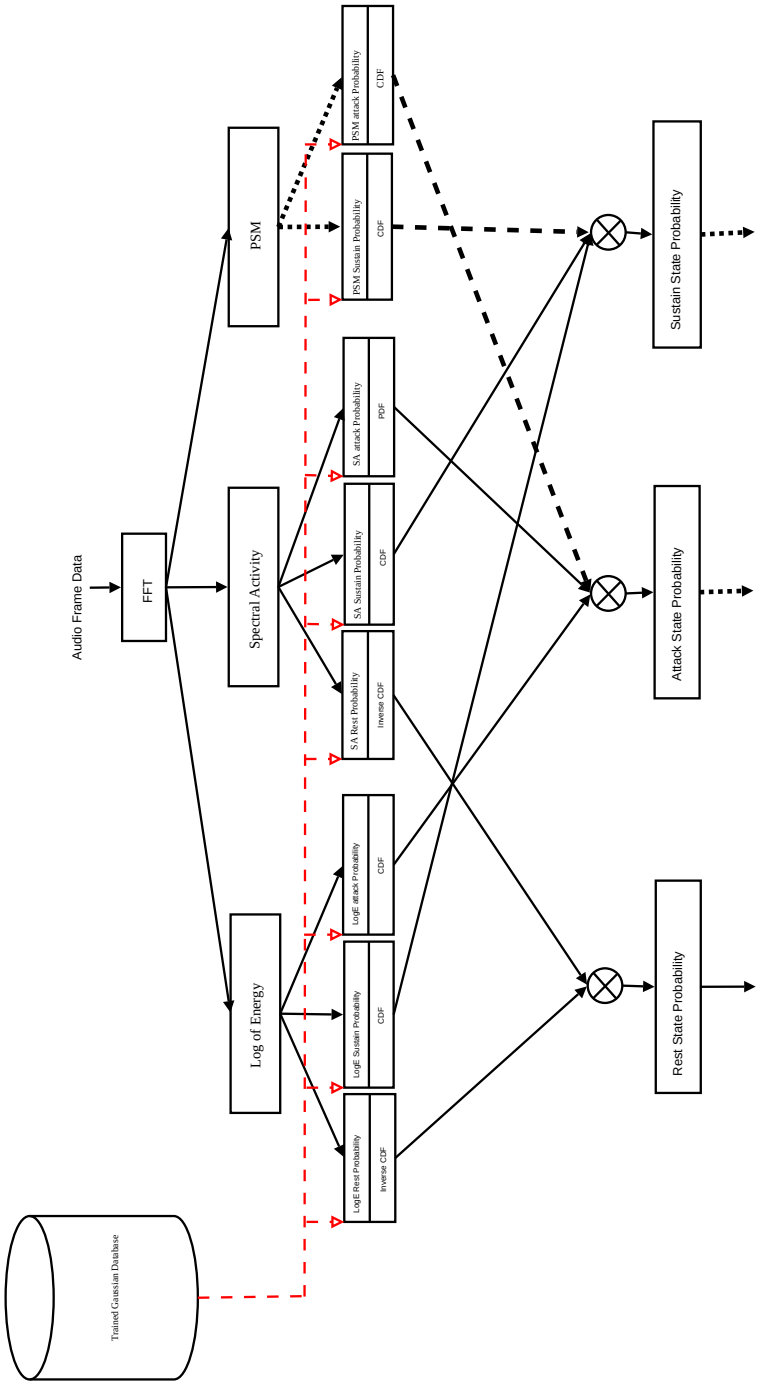

Figure 3.7: New Feature and Probability Observation Diagram (1)

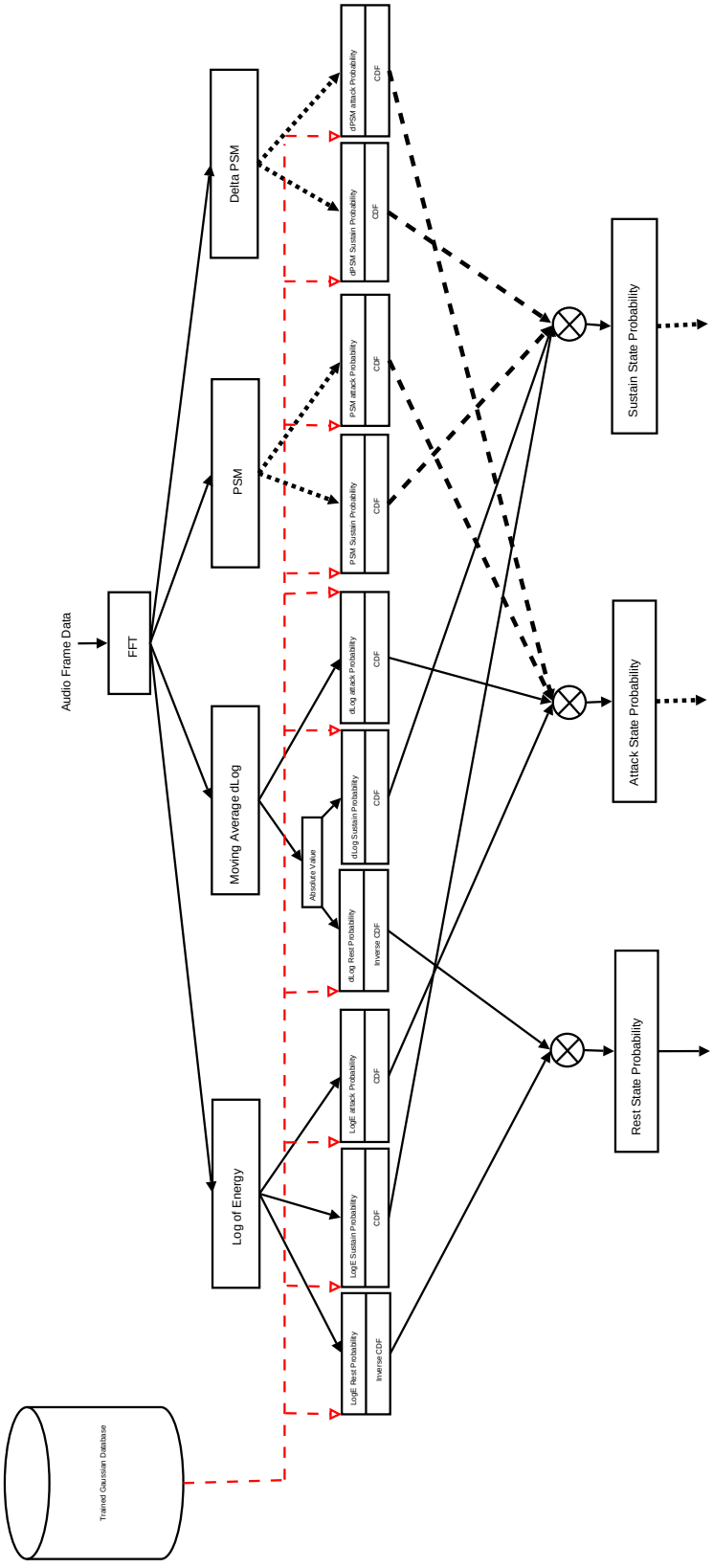

Figure 3.8: New Feature and Probability Observation Diagram (2)

Chapter 4

Training

The principles of form will be our only constant connection with the past. Although the great form of the future will not be as it was in the past . . . , it will be related to these as they are to each other: through the principle of organization or man's common ability to think. — John Cage, *Credo*

In this chapter, we introduce a training method for *IRCAM*'s score following based on designs in Chapter 3. Before we enter the training itself, we need to define the notion of training and also contemplate and differentiate between and ideal training and training in the context of music practice, defining the objectives of training for the score following in question.

4.1 Training and Music Tradition

As discussed before, by training we aim to adapt the *observation* parameters to a certain piece. 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 our case.

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.

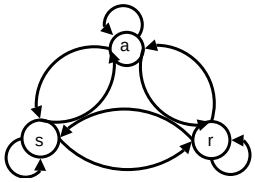

Figure 4.1: HMM for "Orio training"

- 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 run-throughs 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.

4.2 Previous Work

The training method discussed here is proposed by Nicola Orio and Diemo Schwarz and marks the beginning of this project. Note that this method was proposed for the old *observation* block in Figure 1.4 on Page 15.

In their unsupervised training, Orio has proposed using a simple three-state *HMM*, demonstrated in Figure 4.1, with the same transition probabilities and *observation* block as the score following. The system would observe and decode the *HMM* using *Viterbi* algorithm at each iteration and the new μ and σ parameters for each exponential probability mapping is found by a curve fitting on the histogram of the feature observation. Figure 4.2 shows one iteration of this training for Log of Energy's sustain feature where the red-line indicated the newly fitted exponential mapping to be used in the next iteration.

Without going into the evaluation of the score following using the trained parameters, which were not satisfactory, we can criticize this methodology from different point of views:

- This training method assumes the same heuristics criticized in Section 2.1.2 and therefore fails to construct a basic correspondence between the parameters and high-level states.

- Assuming the training is appropriate, this method experience *error propagation*, that is, if there is a tiny error at one iteration by propagation through other iterations it becomes bigger and bigger.

- The curve fitting method is highly dependent on the precision of the histogram constructed over features which is not fixed for different pieces and features and would highly affect the parameters found.

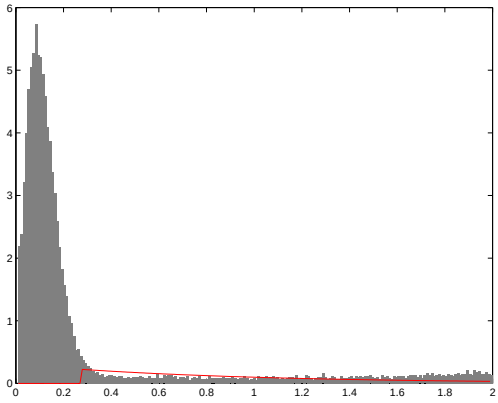

Figure 4.2: One iteration of "Orio training"

- There is no discrimination done for high-level states and therefore we can never be sure whether what we are observing at each iteration correctly corresponds to the desired high-level state.
- It is assumed that this method converges to stable values after several iterations. There is no method in evaluating this convergence and the converged values are highly dependent upon histograms' precisions and starting points and finally, they is no direct correspondence between these values and desired parameters for a better alignment.

Following the above critics and the new *probability observation* constructed in Chapter 3, we propose a training method which is again automatic (if not unsupervised) but more controllable in terms of behavior and intermediate parameters.

4.3 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 (Renals et al. 1993). 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 Cheveigné and Kawahara 2002) has been chosen as this algorithm to provide discrimination knowledge after several considerations.

Figure 4.3 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*.

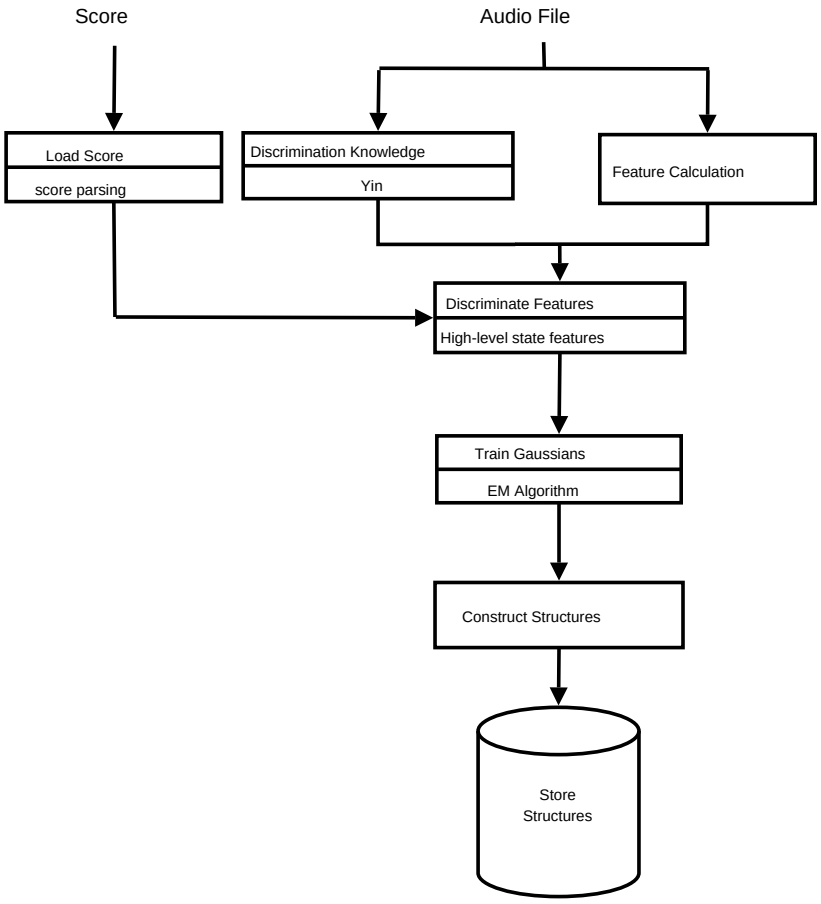

Figure 4.3: Automatic Discriminative Training Diagram

4.3.1 Discrimination

During discrimination, using the discrimination knowledge (provided by *Yin*), the score information (list of all notes in the score) and audio features, high-level state features will be discriminated from each other. For example, the *Spectral Balance* feature will be discriminated to *Spectral Balance - Attacks*, *Spectral Balance - Sustains* and *Spectral Balance - Rests*. In this manner, we would be ready to model the class boundaries and learn the distinction between classes.

The discrimination done here is basically what has been described in Section 3.3 and Section 3.4.3.

4.3.2 Training

Having all high-level state features discriminated, we are ready to model the *gaussians* described in Section 3.4. We evade using fitting algorithms due to robustness and since we are dealing with *gaussian mixtures* (Reynolds 1995; Cappé 2000) and are projecting more mixtures in a row for future, we use *EM Algorithm* (Bilmes 1998; Dempster et al. 1977) to construct the *gaussians* on observed discriminated features.

The result of the training is a set of *PDF*s that correspond to each high-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 (as described in Section 3.4) 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 Figures 3.7 and 3.8.

4.4 Some results and remarks

To train the probability mappings for Figures 3.7 and 3.8, there will be a total of 13 *PDFs* to train. Here we demonstrate for the case of *Log of Energy* feature, the results of each training step as well as the final mapping results for the case of the first movement of *En Echo*, "Riviere," for soprano and electronics.

Figure 4.4 on Page 46 shows the three high-level states (attack, sustain and rest) extracted from *Log of Energy* on the left with the trained *gaussian* in red. Associated with each *gaussian* is the probability mappings on the right which will be stored in the database.

Note that the *automatic discriminative training* implemented is independent of the features. This characteristics is crucial for the development of score following because, as seen before, the notion of score following and expectations of the system evolves with time and necessitates introduction of new features. In this manner, if the new feature has enough "sense" for discrimination, the training will be done in the same manner as before.

One nice considerations for the future of score following is the ability to have local models for each event in the score. With the new system this is totally imaginable and we only need to consider a chain of *gaussians* for each high-level states each corresponding to an event in score. However, such a progress requires a sufficient database of aligned audio for training and we can't rely on one run-through of a rehearsal. This topic will be more elaborated in the concluding chapter.

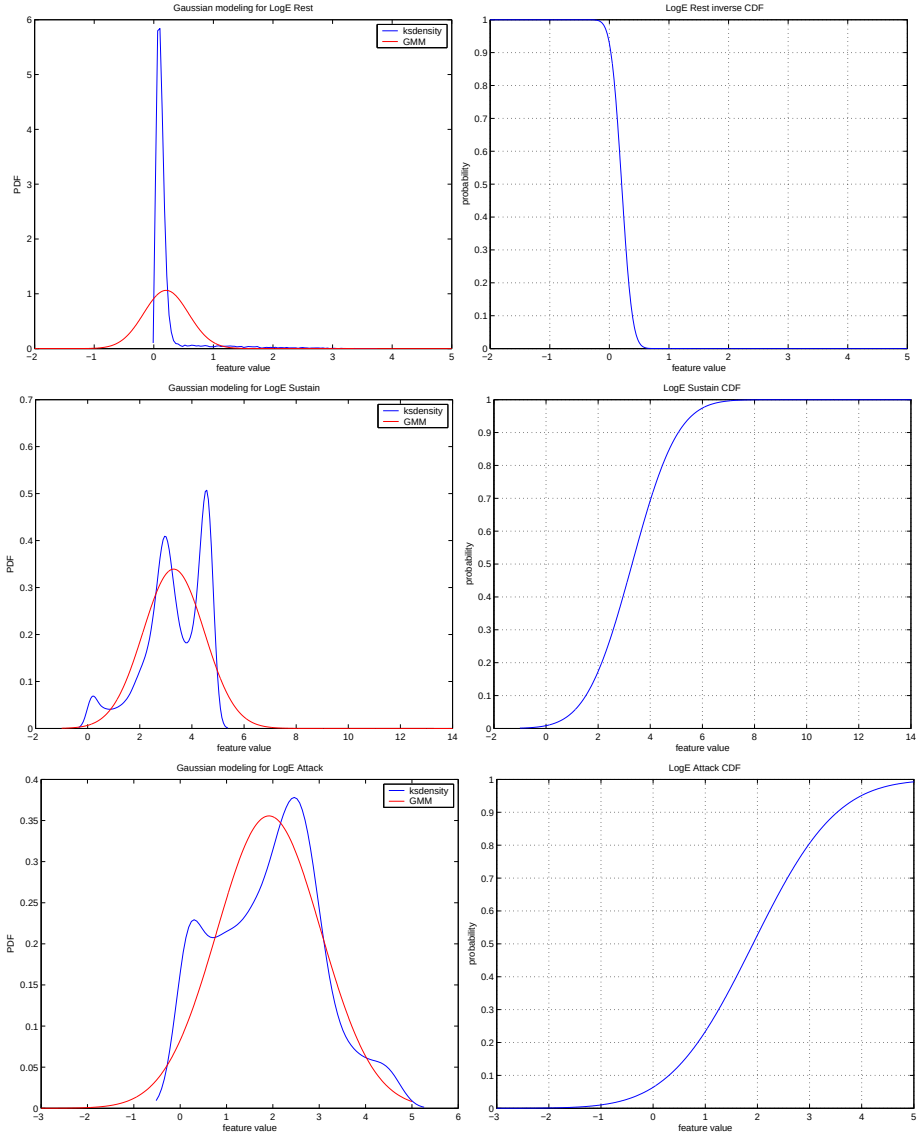

Figure 4.4: Results of training for LogE feature in "Rivier"

Chapter 5

System Evaluation

In this chapter we aim to provide a subjective evaluation of systems in Figure 1.4 (old system), 3.7 (new probability observation with moving average $\Delta\log$ feature) and 3.8 (new probability observation with Spectral Balance feature), with the two latter using the outcome of a training session. The evaluations shown here are simulations of the score follower on *Matlab* ran on audio files. However, we tested the system in several test sessions in studios with Philippe Manoury, Serge Lemouton along with soprano Valerie Philippin on *En Echo* which we will describe.

For the simulation, we limit our observations on segmentation output of the score follower, being the main objective of the score following. We demonstrate score following results on short fragments of music which were problematic in the old system and comment on the output. The examples demonstrated here are excerpts from Manoury's first movement of *En Echo*, "Riviere," for soprano and electronics.

Figure 5.1 demonstrates real-time alignment results for measure 39 and measures $29 - 32$ for the three systems mentioned. Observing from the previous system to the system using spectral balance as feature (system (2) in figure), we can contemplate on the following remarks:

- The stability of the system has increased, in the sense that we do not observe jumps to other states during sustains.
- At some instants, specially fast transitions, alignment is improved.
- Even if a high-level state is not recognized, due to the new system design, the system's behavior can be explained and remarks can be gathered for further development of the system (example: first repeated note in measure 39 in system (2) which is not recognized and is because of higher precision needed in the Spectral Balance feature and finer training.

The real-time tests in the studio, led to similar improvements specially for the mentioned measures and similar instants in the score where we were not able to detect before. Although the score follower turned out to be almost perfect on audio recordings of a performance, its performance in real-time was not perfect to that extent. But due to our design approach, every shortcoming of

the system encountered a direct physical interpretation which could be overcome by adjusting the parameters during performance; thus, leading to more considerations for future research.

As a conclusion for evaluation, we would like to refer the reader to the general contemplations made on the topic of *Evaluations* of score following in Section 1.2.5. An important need for the evaluation of new features and elements as well as final system's output evaluation is a database of aligned music files. This topic will be addressed in Chapter 6.

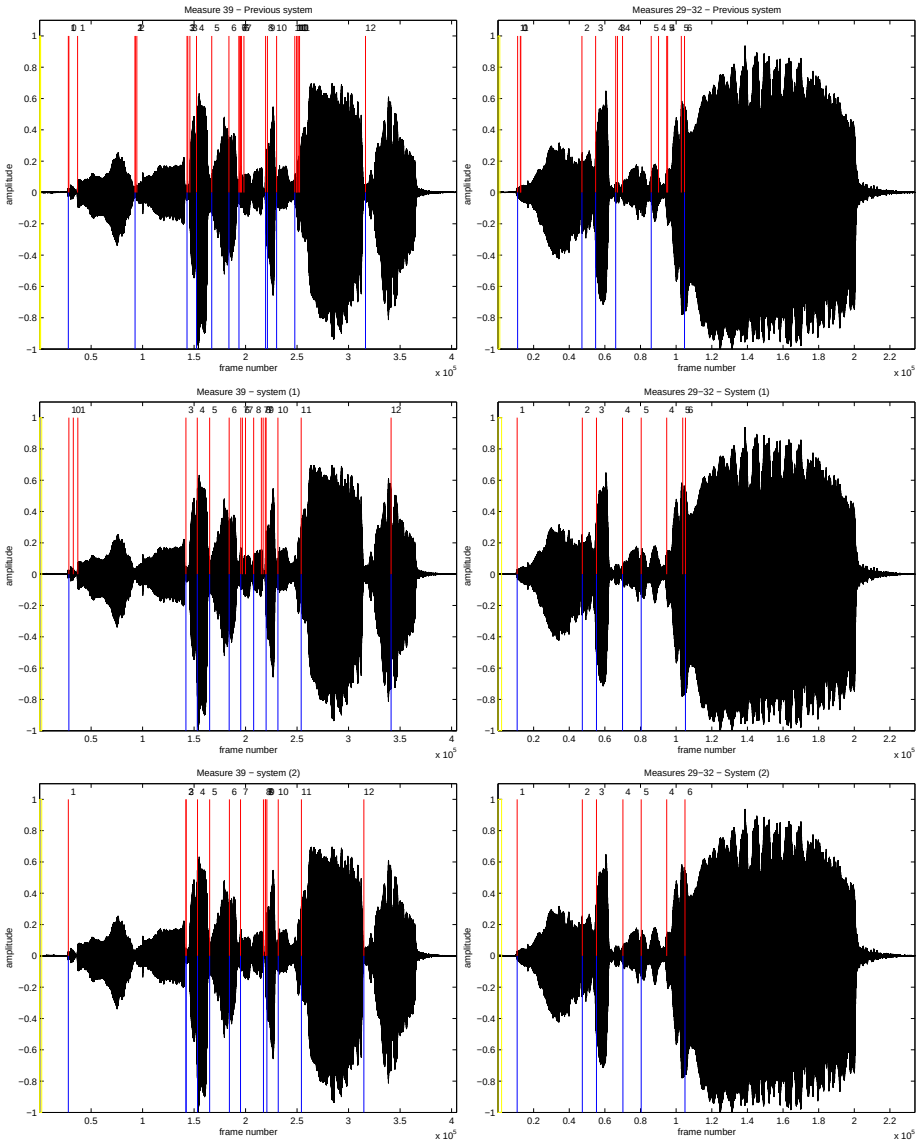

Figure 5.1: Evaluation of the three systems on measure 39 (left) and measures 29-32 (right) of "Riviere." System (1) refers to the observation probability with moving average Δ features; System (2) refers to the observation probability with spectral activity feature and no Δ features.

Chapter 6

Future Works and Conclusion

One day down at Black Mountain, David Tudor was eating his lunch. A student came over to his table and began asking him questions. David Tudor went on eating his lunch. The student kept asking questions. Finally David Tudor looked at him and said, "If you don't know, why do you ask?"
— John Cage, *Indeterminacy*

6.1 Future Works and Remarks

In this section we summarize our contemplations on future works and considerations of score following research. In this manner, we go through each aspect separately since each define separate issues but at the same time correlated towards further progress of score following in general.

6.1.1 The urge of an aligned database of music

Without hesitation, we emphasize on the importance of the availability of a database of aligned music files for further progress of score following research. Since we are dealing with a research oriented towards a *empirical-synthetical* approach, it is important to evaluate the changes made in the system on a large database of aligned music to compare the output and see whether it would be good enough to be used for score following in practice. While small changes and improvements are possible without such database, it limits the research progress since for many proposed works we would need evaluations to ensure the robustness of the system as well as training.

We note at this point that the speech recognition literature was standing at the same point in early 80s and important improvements in their systems took place by having huge databases of speech available for training and evaluation of competing systems.

6.1.2 Towards localized probability modeling

One comment on further development of our system is to move from fixed, "out of time" observations to localized observations. At this moment, the system assumes fixed probability mappings for all high-level states in a score (for example, one probability mapping for all attacks). This assumption again is based on *heuristics* which has never been evaluated since we need the mentioned database. Another approach, which seems reasonable after our experience, is to assume localized probability mappings for each high-level states in the system. With the current system, this means using a *mixture of gaussians* for each high-level state and is completely implementable within our new framework. However, this approach requires the mentioned database for training and evaluations and considerations of such approach for practical purposes.

6.1.3 Feature tests

Furthermore, the modified features along with the new suggested feature should be tested for *robustness*. For this reason, the features need to be run on a large database of sound files and their behavior should be studied which was out of the context of this project.

6.1.4 Temporal Considerations in HMM

We believe that with a strong *observation*, we would always pass to the right state. However, a classical problem with the *HMM* approach lies in its temporal modeling. While this topic was not an objective of this project, we contemplated on it since it would lead to improvements of the general systems. There has been many approaches in the literature towards solving this classical problem such as using parallel *HMMs* and coupling the system with other approaches. As a general remark, it seems that in the context of score following where we try to model a musical (or virtual) time, the idea of using *clock time* (i.e. frame by frame analysis and decoding to *HMM*) is not a good idea. One famous approach which is widely used in music information retrieval is to drop the *clock time* and use the onset information as the time triggers of the temporal analysis (Takeda et al. 2003; Takeda et al. 2002).

6.1.5 Refining the Music model

Furthermore, in the *HMM Music Model* we need to consider adding a *release* model. At the moment, the release model used is just a rest model added at the end of a note model. With the designed observation block and training method, it is totally possible to extract release features from low-level features and train them for detection. This will help for faster and easier recognition of attacks, more cues for special cases such as repeated notes where most features are not

changing and also better transition modeling for the cases of *legato* and *staccato* notes.

6.1.6 Calibration for real-time score following

In order to have satisfying results as shown in evaluation during a live performance, we need to make sure that the feature values produced lie in the same range as the training samples. In the training literature, this goal is achieved by an automatic (if not unsupervised) on-line learning. However, due to practical purposes in a musical situation, this is not possible. One practical solution to this, after our contemplations is to calibrate the solo input of the system before each live performance which is a simple audio engineering task.

For this purpose, we have included range of each feature in the saved structure after training. The audio engineer can ask the soloist to perform a single *fortissimo* note while looking at one or several feature values and by controlling the tabs on the mixer desk, he can calibrate the values to an approximate value near the maximum of the feature range.

This way, we would make sure that during live performance, feature values will occur in the same range used during training which in terms assures desired alignment of real-time score follower.

6.2 Conclusion

The early objective of this project was to implement a training algorithm for the existing score follower at *IRCAM*. Such task requires profound understanding of system's behavior from a control aspect. The project started by studying *IRCAM*'s score following and other systems available in the literature. Throughout this process and after several unsuccessful attempts, the author realized that the incapabilities of the score follower in question lies in the concepts used for the component designs and especially in the *observation block*.

From this moment on, studies and experiments were undertaken to redesign the *observation process* of the score follower which resulted in the system introduced in Chapter 3. The proposed system adds more flexibility in terms of development and physical observation of musical phenomena in score following. In this design, the author emphasizes on critical thinking developed on the basis of use of *heuristics* in the previous systems from a control aspect.

One major outcome of this new design is an implicit notion of parameter control which eases the process of training introduced in Chapter 4. In this respect, a new approach to training called *automatic discriminative training* is introduced which emphasizes on the alignment goal of score following, being the main objective of the system, rather than adapting a perfect model. Moreover, the proposed method is independent of features which makes it usable and flexible for future developments of score following.

While having adapted parameters out of training and through evaluations of the system on several music samples, the author realized that the second stage towards more precision and perfection in score following is refinement of the *features* used during *observations*. In this manner, every feature was analyzed and some refinements as well as new features were introduced as described in Chapter 2. The newly introduced *Spectral Balance* feature, being much more

correlated to high-level states, is currently in use in the score following and has replaced two features in the previous system and has proved to be practical for real-time performance of the system.

Some evaluations of the outputs of the proposed systems were presented in Chapter 5 which shows more stability, precision and specially more control over physical behavior of the new system. Real-time tests of the system have proven improved results and more control over the behavior of the system, as well as introducing a way to go for further refinements of score follower.

Moreover, during this work and facing new designs and challenges in the field the author has gathered several contemplations regarding the future developments of score following in general which was presented in the previous section.

The field of score following, having almost 20 years of age, has recently gained much attention and feedback from other domains specially the music information retrieval literature. Saying this and considering recent advancements in fast computations, we can argue that the rapid growth of this field in applications and development is not quite astonishing and we anticipate more interesting advancements in its technologies and practical aspects in near future.

Bibliography

Baird, B., D. Blevins, and N. Zahler (1990). The Artificially Intelligent Computer Performer: The Second Generation. In *Interface – Journal of New Music Research*, Number 19, pp. 197–204.

Baird, B., D. Blevins, and N. Zahler (1993). Artificial Intelligence and Music: Implementing an Interactive Computer Performer. *Computer Music Journal 17*(2), 73–79.

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

Bloch, J. and Roger B. Dannenberg (1985). Real-Time Accompaniment of Polyphonic Keyboard Performance. In *Proceedings of the ICMC*, pp. 279–290.

Brown, J.C. and Miller Puckette (1992). An efficient algorithm for the Calculation of a Constant Q Transform. In *Acoust. Soc. Am.*, pp. 2698–2701.

Brown, J.C. and Miller Puckette (1993). A high resolution fundamental frequency determination based on phase changes of the Fourier transform. In *Acoust. Soc. Am.*, pp. 662–667.

Cappé, Olivier (2000, June). Modèles de mélange et modèles de markov cachés pour le traitement automatique de la parole. Web page. http://www.tsi.enst.fr/ cappe/cours/tap.pdf.

Castel, Yohan (1999). Human behaviors and psychobiology. Web page. http://www.psychobiology.org/.

Dannenberg, Roger (1988). Method and apparatus for providing coordinated accompaniment for a performance. US Patent No.4745836.

Dannenberg, Roger (2004, June). Private conversation with the author. Hamamatsu, Japan.

Dannenberg, Roger B. (1984). An On-Line Algorithm for Real-Time Accompaniment. In *Proceedings of the ICMC*, pp. 193–198.

Dannenberg, Roger B. and B. Mont-Reynaud (1987). Following an Improvisation in Real Time. In *Proceedings of the ICMC*, pp. 241–248.

Dannenberg, Roger B. and Mukaino (1988). New Techniques for Enhanced Quality of Computer Accompaniment. In *Proceedings of the ICMC*, pp. 243–249.

de Cheveigné, Alain and H. Kawahara (2002). YIN, a fundamental frequency estimator for speech and music. *J. Acoust. Soc. Am. 111*, 1917–1930.

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

Grossberg, Stephen (1982). *Studies of Mind and Brain*, Chapter Learning by Neural Networks, pp. 65–156. D. Reidel Publishing, Boston, MA.

Grubb, Lorin and Roger Dannenberg (1997a). System and method for stochastic score following. US Patent No.5913259.

Grubb, Lorin and Roger B. Dannenberg (1997b). A Stochastic Method of Tracking a Vocal Performer. In *Proceedings of the ICMC*, pp. 301–308.

Loscos, A., P. Cano, and J. Bonada (1999a). Low-Delay Singing Voice Alignment to Text. In *Proceedings of the ICMC*.

Loscos, A., P. Cano, and J. Bonada (1999b). Score-Performance Matching using HMMs. In *Proceedings of the ICMC*, pp. 441–444.

Mouillet, Vincent (2001). Prise en compte des informations temporelles dans les modèles de markov cachés. Rapport de stage, IRCAM.

Orio, Nicola and F. Déchelle (2001). Score Following Using Spectral Analysis and Hidden Markov Models. In *Proceedings of the ICMC*, Havana, Cuba.

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

Orio, Nicola and Diemo Schwarz (2001). Alignment of Monophonic and Polyphonic Music to a Score. In *Proceedings of the ICMC*, Havana, Cuba.

Pardo, Bryan and William Birmingham (2001). Following a musical performance from a partially specified score. In *IEEE Multimedia Technology Applications Conference, Irvine, CA*.

Pardo, Bryan and William Birmingham (2002). Improved Score Following for Acoustic Performances.

Pollard, David (2001). Paris lectures on le cam theory. Web page. http://www.stat.yale.edu/ pollard/Paris2001/lectures.html.

Puckette, Miller (1990). EXPLODE: A User Interface for Sequencing and Score Following. In *Proceedings of the ICMC*, pp. 259–261.

Puckette, Miller (1995). Score Following Using the Sung Voice. In *Proceedings of the ICMC*, pp. 199–200.

Puckette, Miller and Cort Lippe (1992). Score Following in Practice. In *Proceedings of the ICMC*, pp. 182–185.

Rabiner, L.R. (1989). A tutorial on hidden markov models and selected applications in speech recognition. *Proceedings of the IEEE 77*(2), 257–285.

Raphael, Christopher (1999a). A Probabilistic Expert System for Automatic Musical Accompaniment. *Jour. of Comp. and Graph. Stats 10*(3), 487–512.

Raphael, Christopher (1999b). Automatic Segmentation of Acoustic Musical Signals Using Hidden Markov Models. *IEEE Transactions on Pattern Analysis and Machine Intelligence 21*(4), 360–370.

Raphael, Christopher (2001). A Bayesian network for Real-time Musical Accompaniment. *Neural Information Processing Systems (NIPS) 14*.

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

Reynolds, Douglas A. (1995, August). Speaker identification and verification using gaussian mixture speaker models. In *Speech Communication*, Volume 17, pp. 91–108.

Schreck Ensemble and Pieter Suurmond (2001). ComParser. Web page. http://www.hku.nl/ pieter/SOFT/CMP/.

Takeda, Haruto, Takuya Nishimoto, and Shigeki Sagayama (2003, October). Automatic Rhythm Transcription from Multiphonic MIDI Signals. In *4th International Conference on Music Information Retrieval*.

Takeda, Haruto, Nakai Saito, Tomoshi Otsuki, Mituru Nakai, Hiroshi Shimodaira, and Shigeki Sagayama (2002, December). Hidden Markov Model for Automatic Transcription of MIDI Signals. In *IEEE Workhop on Multimedia Signal Processing*.

Vantomme, Jason D. (1990). Score Following by Temporal Patterns. *Computer Music Journal 19*(3), 50–59.

Vercoe, Barry (1984). The Synthetic Performer in the Context of Live Performance. In *Proceedings of the ICMC*, pp. 199–200.

Vercoe, Barry and Miller Puckette (1985). Synthetic Rehearsal: Training the Synthetic Performer. In *Proceedings of the ICMC*, pp. 275–278.