Towards the Automated Analysis of Simple Polyphonic Music: A Knowledge-based Approach

Juan Pablo Bello Correa
Department of Electronic Engineering,
Queen Mary, University of London

Thesis submitted in partial fulfillment
of the requirements of the University of London
for the degree of Doctor of Philosophy

January, 2003

Abstract

Music understanding is a process closely related to the knowledge and experience of the listener. The amount of knowledge required is relative to the complexity of the task in hand.

This dissertation is concerned with the problem of automatically decomposing musical signals into a score-like representation. It proposes that, as with humans, an automatic system requires knowledge about the signal and its expected behaviour to correctly analyse music.

The proposed system uses the blackboard architecture to combine the use of knowledge with data provided by the bottom-up processing of the signal's information. Methods are proposed for the estimation of pitches, onset times and durations of notes in simple polyphonic music.

A method for onset detection is presented. It provides an alternative to conventional energy-based algorithms by using phase information. Statistical analysis is used to create a detection function that evaluates the expected behaviour of the signal regarding onsets.

Two methods for multi-pitch estimation are introduced. The first concentrates on the grouping of harmonic information in the frequency-domain. Its performance and limitations emphasise the case for the use of high-level knowledge.

This knowledge, in the form of the individual waveforms of a single instrument, is used in the second proposed approach. The method is based on a time-domain linear additive model and it presents an alternative to common frequency-domain approaches.

Results are presented and discussed for all methods, showing that, if reliably generated, the use of knowledge can significantly improve the quality of the analysis.

A mi casa, la que no tiene ni paredes ni techo,
sino a Salvador, Maritza, Jesús y la Mava.

Acknowledgments

Completing my doctoral research in a foreign country and in a language which is not my own has been quite an experience, one that can be moderately described as "intense". This dissertation has only been possible given the immense support and cooperation that I received from family, friends and colleagues. For them, to whom I am so profoundly indebted, I dedicate the following remarks.

First of all, I would like to thank my supervisor Prof. Mark Sandler for opening the doors to a life-changing experience. For having the courage of recruiting me when none of my words made any sense. For the continuous practical and emotional support. For the good advise during the length of this project.

Thanks to the Joint Information Systems Committee (JISC) in the United Kingdom, the National Science Foundation (NSF) in the United States and the Fundación Gran Mariscal de Ayacucho in Venezuela for funding my research and for covering my living expenses in London during the last three years.

To the OMRAS people: Tim Crawford, Don Byrd, José Montalvo and Matthew Dovey, for providing the practical context, both academically and economically speaking, for this research to be pursued. To Dr. Mark Plumbley and Dr. Mike Davies for the constructive comments and ideas and for helping to reunite a very bright and enjoyable group. To Dr. Richard Kronland-Martinet and Mr. Thierry Voinier at LMA in Marseille for making available their recording facilities. To Giuliano Monti, for the technical help, for the practicality, for introducing me to the cult of Guinness and, most importantly, for the best zampone in town. To Laurent Daudet, for his brilliant contribution to pitch estimation algorithms, for teaching me the art of writing a paper (an art that was ultimately applied to this thesis), and for a somehow suspicious haggis. To Jeremy Pickens for the work in music information retrieval and for the dodgiest salsa dancing I have ever seen. To Chris Duxbury, for the ideas on onset detection, the introduction to the "real" English language and for not letting me sleep before conference presentations. To Nick Mitianoudis and Jean-Julien Aucouturier for the useful suggestions and discussions and for Vrisaki and the Bourvil samples respectively. To Ian, Chris L., Vangelis, Josh, Samer, Paul, Dawn and all the people, first at King's and now at Queen Mary, that helped to make this such a fantastic experience. To Elina Middleton-Lajudie for the patient English corrections (I wish you could fix my accent as well).

Finally, I would like to thank my friends, and especially, my family, for keeping me alive with your encouragement and love.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

The automatisation of human tasks, from manufacturing to image analysis, is an on-going process, boosted by the steep increase in the processing power of computers and the needs of a society whose economy is based on mass production, distribution and consumption. Therefore, it is no wonder that using computers to recreate music understanding is a growing and active research field with a history spanning over several decades.

However, even for tasks that are trivial for humans (e.g. beat following), using computers to analyse and understand music has proven very difficult and not extremely successful over the years. This is all the more disappointing given the great success in related fields, such as speech understanding. The origins of this situation can be identified in music itself.

Music is a popular art, probably the most popular of them all. People are exposed to it constantly during their everyday life. It is widely accessible. Music is on radio and TV, in supermarkets, in high streets, on the internet, in church. Music exists in all cultures, in all languages and in no language at all. We have been exposed to music since we were born (and sometimes even before then) and we will certainly be exposed to it until our last day. It is not possible to shut your ears to music. It has been a defining part of mankind throughout history.

Yet impressively, a precise definition of music remains elusive. Some

might point out that arts are not prone to definitions in general, as with many other fundamental manifestations and emotions of humanity. However, the accessibility and popularity of music have created the need for its analysis, and analysis is a process that passes through the understanding of what is about to be analysed.

Formally, music has been defined "as the science or art of ordering tones or sounds in succession, in combination, and in temporal relationships to produce a composition having unity and continuity" [Mer02]; or as "the art of arranging sounds in time so as to produce a continuous, unified, and evocative composition, as through melody, harmony, rhythm, and timbre" [AmH00]. For the sake of simplicity, and much to the annoyance of the late John Cage's followers (given his concept-breaking piece *4'33"*), we will embrace such definitions.

There are three main factors common to both definitions. First of all there is the intention, the "ordering" or "arranging" action. Music is, strictly speaking, no random process. With regards to this, it differentiates itself from other phenomena within the broader concept of sound. There is an intention behind music, that of the composer who is assigning meaning to sound events. Intention is a pre-condition to music.

The second factor is the "language", the semantics of music: the "succession" or "combinations" of sounds in "temporal relationships", "the melody, harmony, rhythm, and the timbre" (as instrument selection is also part of the composition). All tools and structures that can be used by the composer to create the music experience, or at least, that can be used by the analyst to decompose the whole into its constituent parts. The combinations are infinite and the rules that bind them, when they exist, are often ambiguous. This complicates the process of music understanding and differentiates it from speech understanding, where rules are tight and combinations are limited.

The third and final factor, but probably the most important, is the perception. Music is there to be perceived, to be listened to. This listening

act, that goes far beyond the physical action of processing an air pressure disturbance, assigns the sensations of unity, continuity and "evocativity" that the definitions mentioned. They do not exist by themselves in the music signal. Music causes reactions in the listener that have to do, not only with the music signal contents, but also with what is in the listener's mind, his/her experience, his/her emotions, his/her knowledge. The listener has an active role in the music process.

Here lies the biggest complexity of analysing music with computers: making algorithmic routines become active listeners. The more complex the task, the more "active" the listener needs to be: we can all follow the beat by tapping our fingers, and we can easily identify the melody on a music track, however, calculating the metric of a song, or identifying the quality of a chord are tasks that require specialised knowledge, and vast experience in similar tasks, thus they are limited to a few experts - e.g. musicians, musicologists - who possess the adequate training.

This is all the more acute with computer systems, considering that they do not have the basic training given by our long-standing, permanent exposure to music in everyday life.

1.1 Objectives and motivations

The process of music transcription aims to convert a musical recording or performance into a musical score. This is only a "linguistic" description of the music, an analysis of music into its components, broadly speaking: melody, harmony, rhythm and timbre.

Except timbre, which is related to the played instruments, the other components have a common denominator: musical notes.

Melodic lines are sequences of notes over time, usually containing salient and repetitive motifs. Harmony is determined by the relationship between the pitch of the notes being played simultaneously and individually during the different parts of a song. Rhythm is determined by the onset time of the notes and the accent with which the notes are played.

Then it seems a reasonable assumption that, if analysing single instrument recordings, the estimation of musical notes provides enough information to describe a musical file (enough at least for a range of high-level applications).

The research in this dissertation aims to develop a system that extracts note events from simple polyphonic audio files, i.e. real recordings including an undetermined number of notes at a time played by a single instrument. We do not pretend this to be a transcription system in the proper musical sense, but a high-level music analysis system. However, and as the term is used to this purpose in the literature, we will indistinctly refer to our results as transcriptions and sometimes to the research area as automatic music transcription.

It is important to note that we do not consider the analysis of single-instrument music to be less complicated than that of multiple instruments. Certain orchestral configurations have defined polyphonies, and the timbral differences between instruments can be used to identify mixtures of notes. On the other hand, notes played simultaneously using single polyphonic instruments such as the piano, are sometimes indistinguishable given the interaction between the different elements within the mechanism of the instrument, and the timbral similarities. They are different problems, and it was decided to tackle only one in order to limit the scope of this project.

There are many possible applications for such an analysis system: Coding of audio information for fast audio transmissions through data channels (e.g. MPEG-4 coders), real-time high-level interaction between musicians and computers, analysis of recordings of the same piece by different performers, etc.

In fact, the work of this dissertation is part of the On-line music recognition and searching (OMRAS) project. The intention is to develop a system able to search large databases of symbolic music data (e.g. scores, MIDI) using real recordings as polyphonic queries. The project, part of the wider field of music information retrieval, also provides solutions to the problem

of classification, indexing and storage of large quantities of recorded music.

1.2 Overview

For our analysis, methods will be proposed to estimate the individual features of a note event: pitch, onset time, duration and energy; and to combine the obtained information to form musical notes. We will propose that as with humans, the use of high-level knowledge increases the chances of a successful estimation. In this dissertation, the organisation of these ideas is presented as follows:

Chapter 2 concentrates on developing a framework for the recreation of the music understanding process. It starts by presenting the longstanding argument between the two predominant approaches to comprehension: bottom-up and top-down processing. It discusses and adopts the view that sees comprehension as a complex interactive process between the straight analysis of sensorial data and the use of hypotheses and expectations derived from previous knowledge. The blackboard system is used for the computerised realisation of the interactive approach. It uses a standard architecture: blackboard database, knowledge sources and scheduler. In the final sections the organisation of these elements, as used for our implementation, is explained.

Chapter 3 reviews a number of time-frequency representations as applied to the analysis of musical signals. An extension of the Fourier transform, the phase vocoder, is presented and favourably compared against multiresolution approaches, perceptual models, bi-linear distributions and parametric techniques. In this chapter, the aim is not to propose new techniques but to justify the choice of a certain representation. This is relevant, as the information provided by such analysis is the ground on which the following stages of the system are built. The presented theory is necessary for the development of the novel ideas presented in chapters 4 and 5.

Chapter 4 proposes the use of phase information for the detection of

note onsets. Initially, attack transients are defined and characterised. It is discussed how these characteristics have been used for the task of note onset detection.

The use of phase information is proposed as an alternative to the common energy-based approach. It provides a robust measure for the non-stationarity introduced by attack transients. Simple statistics are used to quantify these observations. The integration into the blackboard framework is explained and results on real CD recordings are presented and discussed. They show improvement over common energy-based onset detection methods.

Chapter 5 introduces two algorithms for note recognition in polyphonic mixtures. First the basic concepts of pitch, monophonic and polyphonic music are introduced. Our focus on polyphonic pitch estimation is stated. Previous approaches to polyphonic analysis are presented and discussed, highlighting strengths and weaknesses. Then, the main complexities of the problem are discussed.

A first system is proposed that aims to estimate pitches based on frequency domain information. It performs analysis on a frame-by-frame basis, by relating spectral information to target harmonic comb structures. It is shown how this analysis is complemented by grouping and evaluating information across time, before producing the final output of the system. Integration into the blackboard architecture is explained and results are presented on a test-bed of real piano recordings. Results show susceptibility to the common problems of polyphony and harmonicity.

A second, alternative approach is then proposed that aims to estimate pitches from the analysis of the time domain waveform of the signal. It assumes each waveform segment to be the weighted sum of the waveforms of individuals notes of the played instrument. In this context, the estimation task is reduced to the calculation of the weighting vector. The individual waveforms form a database that stands for high-level knowledge about the music being played. The problems, caused by the phase misalignment be-

tween sound and database, and the generation of an adequate database, are analysed and discussed. Solutions are proposed and a final system is implemented and integrated into the blackboard framework. Results on real recordings are shown and discussed. It is illustrated how the use of high-level knowledge improves results if provided with an adequate database.

Chapter 6 presents the conclusions and suggest possible directions for future research.

Chapter 2

Framework

2.1 Theories of comprehension

We are concerned with developing a computer system that understands, or comprehends, the phenomena of music. However, comprehension is not easily defined. One view [Gib66, Gib79] regards comprehension as the reconstruction of the intended meaning of a communication. This presupposes that the meaning resides in the message awaiting interpretation. A second view [Bru57, Nei67, Gre72, Gre80] sees comprehension as the product of an interchange occurring between message and receiver. Here, meaning depends on the receiver's own thinking process during such interchange, conditioned by its experience and prior knowledge.

This can be more easily visualised by assuming comprehension to be a hierarchical process. Lower levels of the hierarchy are associated with physical stimulus (i.e. sound in our case) while higher levels are related to constructing meaning from such stimulus [Par97]. In this context, the first view supports the idea of a process that starts at the lower levels of the hierarchy and linearly progresses to the top of it. This way of processing is known as bottom-up or data-driven model. The second view stresses the influence of the higher levels of the hierarchy on the processing of the available information from the lower levels. This approach is known as top-

down or prediction-driven model.

In this chapter both models will be initially discussed and compared. The interactive model will be presented as an option that integrates both processing approaches. The blackboard system is introduced as a practical realisation of the interactive model. This is illustrated by reviewing previous blackboard systems used within the context of music understanding. Then, a system is proposed that uses the blackboard paradigm for the task of polyphonic transcription. It provides the framework into which all systems to be presented in the following chapters will be integrated.

2.1.1 Bottom-up processing

The data-driven model assumes that processing starts with small units of information, gathered at lower levels of the comprehension hierarchy. These small units are progressively clustered at the different levels of the hierarchy forming more complex (and larger) units of information. The final units are those that assign meaning to the overall scene being perceived, right at the top of the hierarchy. The theoretical model, also known as the model of *direct perception*, was first proposed by Gibson [Gib66, Gib79].

For the model to be successful, the assumption must be made that there is enough information at the stimuli level for a comprehensive scene to be constructed. It must be noted that, in this approach, higher levels of comprehension cannot be accessed if lower levels are still unresolved.

To use a common example [Par97], under this view a reader first perceives individual letters, organises them into words, words into sentences, etc. All letters in a word must be processed before it can be understood. In the same way, all words in a phrase must be processed before any meaning can be assigned to it. The process is illustrated at the top of Fig. 2.1.

In problem-solving terms, this means that the development of the solution to a certain problem is a direct consequence of the linear evolution of the available data through a sequence of processing modules. This assumption backs the construction of refining architectures for data processing, that

synthesise a solution from its most basic parts.

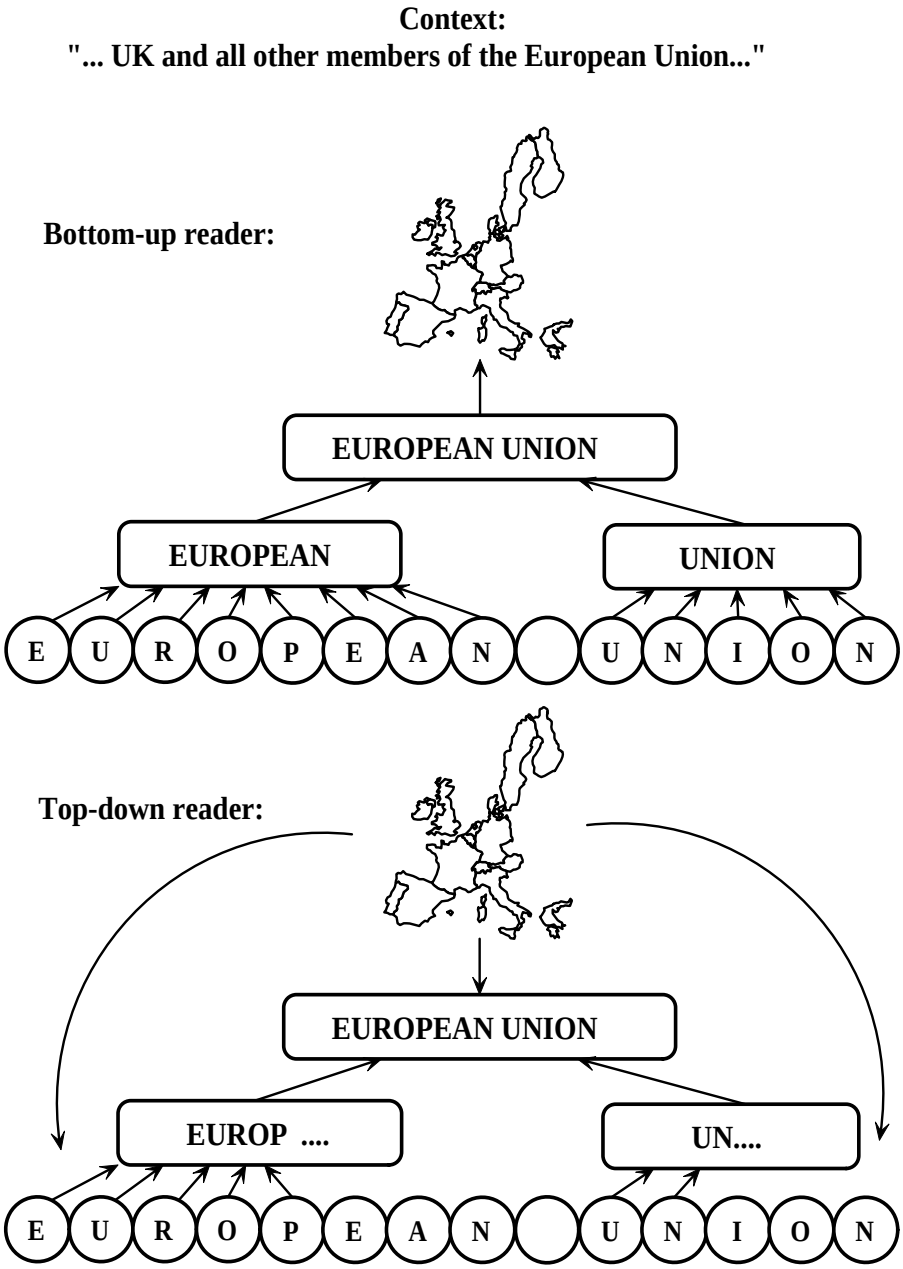

Figure 2.1: Reading using data-driven (top) and prediction-driven (bottom) processing models.

2.1.2 Top-down processing

In contrast, the prediction-driven model proposes that comprehension occurs in a non-linear fashion, starting at the top levels of the hierarchy and selectively using the lower levels to maximise the reception of "meaningful" stimuli. The theory, of *constructive perception*, was introduced by several scientists [Bru57, Nei67, Gre72, Gre80], and is summarised by Eysenck and Keane [EK90] as follows:

1. Perception is an active and constructive process.
2. Perception is the by-product between the presented stimulus and the receiver's expectations, hypotheses and knowledge.
3. Since hypotheses and expectations will sometimes be incorrect, perception is prone to error.

Returning to the previous example, "top-down" readers do not need to identify all letters in a word nor all words in a text to uncover its meaning. They move forward guessing the meaning of words and phrases, only sampling the text to ratify those guesses. This is shown in the bottom of Fig. 2.1.

The approach provides an explanation for comprehension with very distorted inputs, a fact that has been experimentally demonstrated a number of times (see [Bre90] for an account of such experiments in audition). The view supports the problem-solving approach that intends to accumulate experiencing knowledge in order to interpret what is being looked at on a sensory level, and to build up expectations about future stimuli.

2.1.3 Comparison of approaches

After these definitions however, the question of whether perception is a direct or a constructed process remains open. It is safe to say that perception combines both processes at varying levels of importance depending on a number of factors.

Bottom-up advocates are right to claim that sensory input is mostly accurate and that in "normal" conditions it hugely explains perception. However, there are many situations when the direct approach might seem a little oversimplified. If our perception depends purely on the accuracy with which we sense the external world, then there is no plausible explanation for the interpretation of distorted or inaccurate "observations". As constructivists have demonstrated time and time again, even when exposed to very brief and distorted stimuli, subjects are able to correctly interpret test observations.

Furthermore, bottom-up processing fails to address the obvious relation between meaning and previous knowledge (both theoretical and emotional). For example, the reading of this thesis does not mean the same to an engineering student as to a butcher, however, if the butcher happens to be the author's father there is a different, emotional dimension to his reading of these words. None of this can be explained without using the top-down approach.

However, in its pure form, the theory of top-down processing seems to suggest that sensory information is often corrupted. If this is true, then our perception is constantly depending on the use of inference and expectation [EK90]. As perception is correct most of the time, we find ourselves in the compromising position of explaining how inference and expectation are correct most of the time. The explanation is that there is no explanation, because in reality we do not seem to use inference and expectation nearly that much.

We will take the view that perception primarily depends on the use of sensory information while keeping the ever-present capability of using previous knowledge to award meaning to our observations and to effectively modify the subject's focus of attention. This is particularly important when dealing with complex perceptual tasks (i.e. when transcribing music).

Obviously what applies to perception in general, applies also to audition, and the issues that affect human audition are mirrored in the recreation of hearing (and hearing related tasks) using computer systems.

Data-driven systems heavily depend on the un-corrupted digital representation of the signal. This means that, in contrast with human listeners [Bre90, War70], these systems are not able to infer an acceptable answer when a poor image of the original signal is provided. Also, when dealing with complex signals (multiple timbres and musical structures), these systems are incapable of making overall observations of the data [Kla98a] to trigger changes on the low levels of the "perceptual" hierarchy (i.e. focusing on a particular melodic line or rhythm). This inflexibility further complicates the task of automatically obtaining information from the audio signal. However, pure top-down systems cannot substitute analysis based on low-level information, hence a satisfactory solution to the problem of automatic audition lies in the integration of both processing approaches.

2.1.4 The interactive approach

Neisser [Nei76] proposed that perception depends on the complex interaction between bottom-up and top-down processing. This is know as the "interactive-model" approach. The concept is rather general and has been used for different theories regarding specific perceptual tasks. We will draw special attention to the interactive model proposed by Marslen-Wilson and Tyler [MWT80] in the context of word recognition. In their model, different knowledge sources interact in complex ways to successfully analyse language. These knowledge sources comprehensively map the knowledge needed for the understanding of spoken language (i.e. semantic, syntactic, etc). Furthermore, instead of fixed processing steps, the model is flexible enough to allow various processing activities to occur at the same time. In their experiments, it was shown that subjects were faster on identifying pre-defined words in a sentence when context information was meaningful (as opposed to random or incoherent phrases). They concluded that when context did not help, subjects were obliged to process sensory information in detail, hence increasing the response time. Warren and Warren [WW70] found that when subjects were presented with a distorted word within a

slightly-varying phrase, the perception of the word would change according to the new context. These experiments demonstrate the simultaneous interaction of direct and constructive processing.

The translation of these concepts to the context of automatic processing is not immediate. The concepts of knowledge and experience, of easy interpretation under the context of human perception, acquire a different dimension in automatic problem solving.

In this dissertation we will assume bottom-up processing as compatible with the sequential nature of computerised processing systems (given the analogy between digitisation and sensory perception). We will also assume top-down knowledge as the long-term summarisation of bottom-up analysis results (i.e. the training of a neural network, the learning of a note database from the signal itself). Finally, the blackboard framework for processing integration will be used as an implementation of the interactive-model of perception.

2.2 The blackboard framework for processing integration

The blackboard system is a problem-solving model capable of integrating knowledge from different sources and of generating the control information necessary for the interaction of the different parts of the model in an opportunistic and flexible way. It was introduced into AI literature by Newell [New62], as an alternative to the chaining models for expert systems, where data is processed in a fixed, backward or forward, fashion. In their book, Engelmore and Morgan [EM88] use an example to illustrate the model's functioning:

Imagine a group of people trying to solve a jigsaw puzzle, each holding a different number of pieces. At the beginning the most promising of the pieces are attached to a sticky blackboard exposed to the group's view. As the pieces are attached each person independently decides to contribute

if his/her pieces fit the pieces on the blackboard. Note that the problem can be solved without direct communication within the group. Also the development of the solution is independent of the number and kind of pieces each person holds. The solving process is incremental (one piece at a time) and opportunistic (when the opportunity of putting a piece arises).

Note that the model only requires a central database, the sticky blackboard where the puzzle pieces are attached, and a group of experts or knowledge sources, the piece holders, with the ability to interact with the database at will.

Now, suppose that there is limited access to the sticky blackboard. Only one person can be selected to interact with it at a time. If we want to keep experts from communicating, then we require an arbiter or monitor to decide who will attach the next piece. An expert with a relevant piece can raise his/her hand to inform the monitor. Then the monitor can select holders according to the speed at which they raise their hands or the "importance" of their pieces for the solution of the problem.

Before the "limited access" situation, the control of the model lay with the people of the group. In a computerised environment this implies some form of independent processing for each of the knowledge sources. Disposing of a number of processors in a parallel architecture will be a costly solution. Hence, the presence of a monitor, holding strategic and planning responsibilities, is adequate. Having such a module is better suited for single processor environments. This is the approach taken by most blackboard applications.

2.2.1 Components of a blackboard system

The defining parts of a blackboard system, shown in Fig. 2.2, can be enumerated as follows:

1. *The blackboard or global database* is where the problem-solving data is kept. It consists of objects from the solution space (input data, partial or final solutions, etc) hierarchically organised into levels of analysis. It is public in nature, meaning that it is available to all

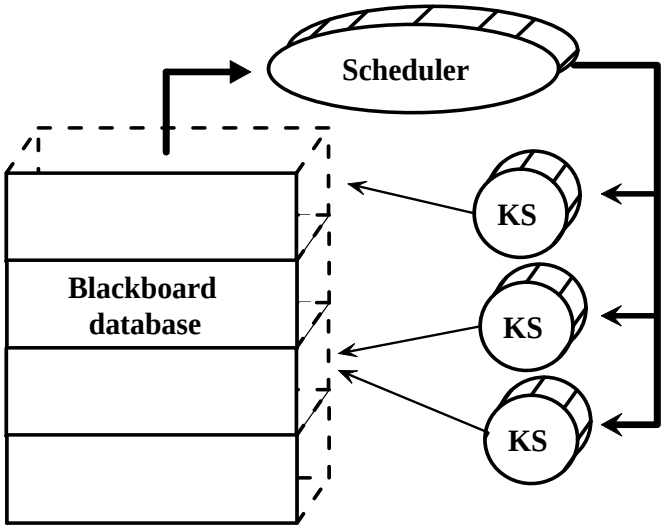

Figure 2.2: Components of a basic blackboard system.

participating modules. Objects in the database can be linked within or between analysis levels, forming hypotheses whose nature could be supportive or competitive. As Ellis [Ell96] notes, the hypotheses on the blackboard depict the state of the analysis process.

2. *The knowledge sources or experts*, are the modules acting upon the information on the database. Their actions correspond to the specialised knowledge they hold and contribute to the solution of the problem facing the system. They are independent of each other. Their only interaction is through the database. Each knowledge source possesses a precondition, or firing condition, that determines under which context it is called into action. When a knowledge source's precondition is satisfied the action contained in its programming body is executed, this is why knowledge sources are usually defined as precondition-action pairs [Mar96b]. The architecture is open regarding the number of knowledge sources that are available to the system. As each expert adds more knowledge to the system, it can be assumed that, if well organised, the performance is improved by increasing their quantity [REFN73].

3. *The scheduler or monitor* is the managing module that controls interactivity within the system. It monitors the changes on the blackboard and decides whose actions to perform next. Its control logic is dynamic, maximising the actions of all other modules at each time step. If well designed, the operation of the system is dynamic and opportunistic [EM88].

2.2.2 Previous approaches

The first known application of a blackboard system, the *Hearsay-II system* [LE77], was developed to be a speech recogniser. Speech recognition is a perceptual problem characterised by a huge solution space, noisy and unreliable data and a large and diverse knowledge-base on which to support a decision. It suited well the interactive approach to problem-solving and the then recently proposed blackboard framework. The resulting system, acknowledged as the "original" blackboard architecture, shaped the development of blackboard systems across a large range of different applications [NI86]. Moreover, the proposed architecture can be (and is being) used as the backbone framework for many auditory perception tasks (speech recognition being one of many tasks of the auditory scene analysis problem [GB99] [Nii86a] [Nii86b]). We will mention some of them.

The Integrated processing and Understanding of signals (IPUS) system was proposed by Lesser et al [LNGK93] [LNK95] for the detection of audible signals in complicated environments. A generic architecture is proposed that concurrently searches for appropriate signal processing algorithms (SPA as defined by the authors) and their optimised set of parameters, depending on the context. IPUS uses the RESUN framework [CL91] to control the knowledge sources' interaction with the database. The focus is on eliminating uncertainty associated with the hypotheses until a likely answer is obtained. Knowledge sources are in charge of resolving the "sources of uncertainty" (SOU) related to each hypothesis. The overall solution is slowly built by setting and reaching smaller sub-goals.

The system maintains the interaction between the processes of selecting the appropriate signal-processing algorithm for the environment, and of interpreting the SPA's output. It monitors the quantities of missing or irrelevant information under the current SPA. This made IPUS a base structure for later signal analysis systems, and in particular for music scene analysis applications.

A system for the automatic transcription of piano music was proposed by Martin [Mar96b] using the blackboard structure developed by Ellis [Ell96]. The framework is also based on the RESUN architecture. It aimed to transcribe four-voice piano versions of Bach chorales. A blackboard database used five hierarchical levels: tracks, partials, notes, intervals and chords. It interacted with a group of thirteen knowledge sources, broadly organised in three different areas: garbage recollection (elimination of wrong supportive or competitive hypotheses), knowledge from physics (that related to spectral behaviour of musical notes) and knowledge from musical practice (i.e. construction and detection of intervals). The latter is an interesting knowledge group, which aims to find intervals and chords of a certain structure. This specificity helps the task of analysing Bach's chorales, while limiting the generality of the system. A second implementation [Mar96a] improved the system's performance by implementing a complex front end based on the log-lag correlogram [Ell96].

The Organised Processing Toward Intelligent Music Scene Analysis (OPTIMA) system proposed by Kashino et al [KM98], aimed to output a score-like representation of a multitimbral musical signal. The system was divided into three main parts: pre-processes, main processes and knowledge sources. The pre-processing stage is concerned with signal analysis, peak-picking and onset detection. The main process block is the blackboard of the system. It is divided according to its information hierarchy in frequency components, notes and chords; and according to the processing of this information in bottom-up, top-down and temporal processing. Blackboard data is integrated by means of Pearl's Bayesian network [Pea86]. The network

propagates diagnostic and causal support for hypotheses in the blackboard represented by a probabilistic grid. Nodes on the grid are connections between successive levels of the hierarchy. The modules acting over such nodes either generate information to be propagated (bottom-up modules) or evaluate that information based on previous knowledge (top-down modules) or past information (temporal modules). The knowledge sources are repositories of rules and data needed by the Bayesian network to integrate the hypotheses information present on the main processing block (i.e. statistical information about chord progressions or note intervals). Note that the use of a Bayesian network for knowledge and data integration eliminates the need for a centralised control module or scheduler. It suggests a sophisticated way of embedding all control information into the blackboard itself. This is an alternative to the RESUN architecture that has been predominant in most blackboard frameworks implemented for tasks within the computational auditory scene analysis problem.

Godsmark and Brown [GB99] introduced a blackboard framework for the automatic grouping of information within an auditory scene. It exploits the context-sensitivity and the retroactivity of auditory organisation. It also uses results from experiments on psychophysical phenomena to regulate the integration of evidence from multiple sources. It proposes an *organisation hypotheses region* (OHR), a temporal window where results are not definitive. Hypotheses surviving analysis inside the OHR, are considered as present organisations within the analysed signal. At the lower levels, hypotheses are grouped primitively (based on bottom-up reasoning), while at higher levels the grouping is driven by schemas, specific pieces of knowledge about complex musical structures (i.e. metre, melodic lines). Low-level information is used to build *synchrony strands* [Coo93] representing dominant spectral components across time and frequency. These *strands* are used in turn to build *streams* [Bre90], more complex forms that develop over long periods of time (i.e. melodic lines, phrases). High-level knowledge includes the grouping of information according to *pitch proximity* and *spectral similarity*.

The system is evaluated on complex identification tasks with synthesised music and its performance compared to that of human listeners. Although success is not clear from the experiments performed, the importance of this system lies in the generality of its framework. The experiments show only an aspect of its possibilities, but as an architecture, it is intended to tackle the whole of the computational auditory scene analysis problem. Its implementation for different problems can be achieved by developing specific sets of knowledge sources.

2.3 Proposed framework

The blackboard framework, as an implementation of the interactive model, suits the computational recreation of complex auditory tasks such as the automatic transcription of music. It comes as no surprise then, that we decided to use this architecture to integrate the complex parts of our system (to be described in the following chapters). Early frameworks were presented by the author in two exploratory systems.

In the first of those systems, a simple blackboard structure was implemented to tackle the problem of monophonic music transcription [BMS00a]. It was fed from a simple analysis stage (short-time Fourier transform plus an energy-based onset detection algorithm) that segmented and retrieved frequency-domain information from the signal. All temporal analysis was constrained by the onset detector. Once the segmentation was performed, all information within the segment was destined to produce a single note starting at the detected onset and finishing at the next onset or offset. The spectral information was then introduced into the blackboard database consisting of a three-levels hierarchy: tracks, partials and notes. The information was grouped and evaluated by simple knowledge-sources until a winning hypothesis was selected (a winner-take-all approach).

The second system aimed at a more ambitious goal: the transcription of polyphonic piano music [BMS00b, BS00]. It featured a more sophisticated architecture that added top-down processing to the pure bottom-up

approach of the previous system. The front-end remained the same, relying on the idea of segmentation as a generator of data for the blackboard. The hierarchy of the database incorporated a new level, that of *chords*, to cope with the complexity of the analysed data. The inclusion of information at the *chords* level was determined by a complex knowledge source, a neural network that estimated the polyphony of the signal. This neural network was trained off-line on a sample test-bed of synthesised piano music. The system carried with it the limitations of a segmentation-based analysis. It also suffered badly from the usual problems of multiple pitch detection (as will be studied in chapter 5). Moreover, the neural network was limited to the detection of polyphonies of up to three notes (given the training space) within a small frequency range, in samples of synthesised piano music.

In any case, it is important to emphasise that more than producing working transcription systems, these algorithms aimed at exploring the concepts of bottom-up and top-down processing and their realisation through the blackboard architecture. As a result of these explorations, an architecture is now proposed that implements the interactive approach in a more comprehensive way. The system's structure adheres to the usual blackboard organisation: blackboard database, knowledge sources and scheduler. Its detailed explanation is provided in the following.

2.3.1 A two-dimensional database

The blackboard database of this implementation uses a two-dimensional internal hierarchy. It contains both spatial and temporal information, corresponding to the multi-dimensionality of musical data. As in any time-frequency analysis, the time-domain waveform is segmented in short sections or frames. However, while some levels feed from data corresponding to the analysis of the current frame analysis, other levels of the hierarchy are generated from the analysis of data belonging to several frames across time. Let us first illustrate the database's structure based on frame information. This is depicted in Fig. 2.3. Arrows correspond to bottom-up processing

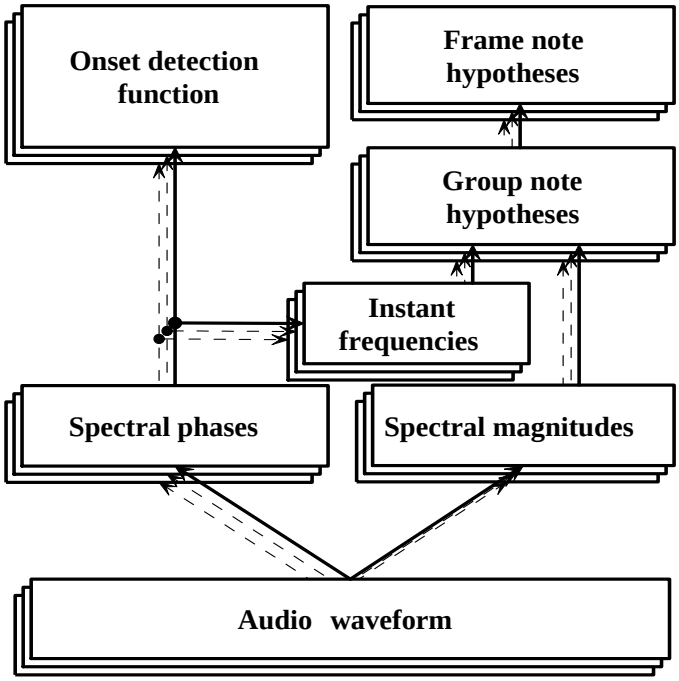

Figure 2.3: Blackboard database in one frame.

paths, emphasising the hierarchical structure.

The lowest data block in the hierarchy corresponds to the segmented audio waveform. This unprocessed data is the starting point for all analysis within the system. Bottom-up and Top-down processes will operate on this data to produce higher levels of information.

The spectral information constitutes the following step in the hierarchy. Magnitude, phases and frequencies from the time-frequency analysis provide the ground information for the rhythmic and harmonic analysis to be performed within the system. The low-level feature extraction that implies the generation of this data is explained in Chapter 3.

The following data blocks are the intermediate levels of the overall hierarchy. They are mid-level representations of the signal, not enough to semantically describe its contents. The "group hypotheses" block contains possible harmonic patterns generated from the spectral information. The "onset detection function" and the "frame note hypotheses" are the instan-

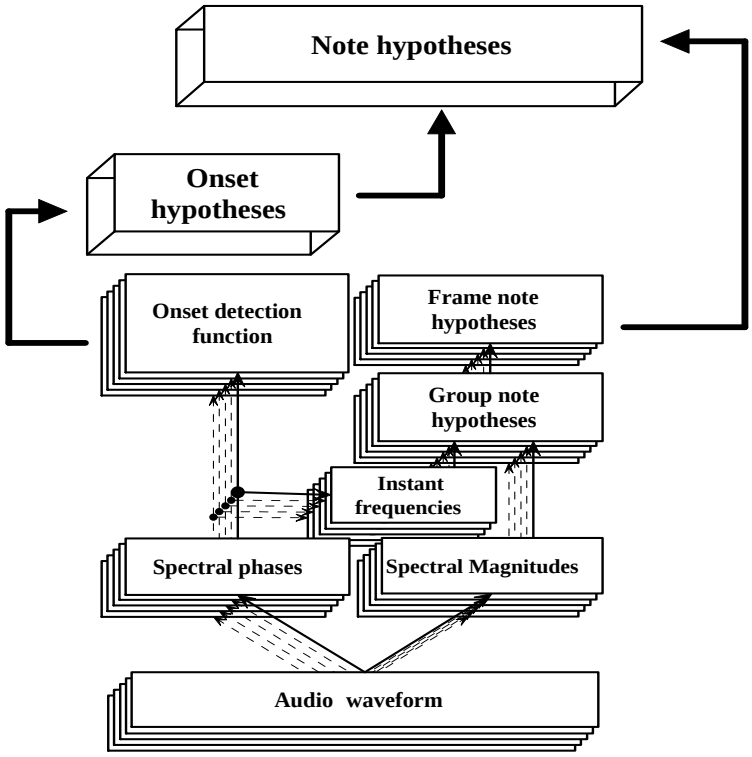

Figure 2.4: Blackboard database over time.

taneous estimations of the system for the musical features in the analysis frame. They serve to generate time-domain functions that are analysed to produce higher levels in the hierarchy.

Note that while lower levels correspond to processing data, intermediate and, as will be seen, higher levels introduce data corresponding to hypotheses and expectations. In these levels, the system uses its knowledge to propose possible solutions to the local observations of the signal. These propositions are not definitive (as processing data is) and not unique, hence creating competition within the system. Their survival depends on the support they are able to gather within the factual levels of the hierarchy.

Music is a highly repetitive temporal process. Only from the consistent observation of events in time can high-level hypotheses be generated. When looking at the signal on a frame-by-frame basis, we are susceptible to draw incorrect conclusions from our observations. Local noise or instability pro-

duces large quantities of spurious values, thus, data from the middle of the hierarchy cannot be used as the system's output.

Higher levels are those that summarise observations across time. In our framework there are two such levels: "onset times" and "note hypotheses" (see Fig. 2.4). The former contains timing information about musical events in the signal. The latter describes musical notes in terms of their pitch, onset time, energy and duration. Surviving hypotheses at the "note hypotheses" level are used to build the output of the system (in the form of a MIDI file).

In this system there is no high-level musical knowledge evaluating the quality of intervals or chords (as it will constrain our solution space). Hence, we deemed it as unnecessary to include those levels into the hierarchy. However, as different note hypotheses are allowed to share onset times, they are implicitly included within our final data block.

The process which generates both intermediate and high levels of the database is the subject of chapters 4 and 5.

2.3.2 Knowledge sources

There are two major roles for knowledge sources in this system: an active role in which they are in charge of reading and modifying the system data according to the specific knowledge they contain; and a passive role in which they keep experiencing knowledge available for active modules to use. This describes a double functionality: processing and storage. However, this functionality only corresponds to an overview of a finer classification, as can be observed in Fig. 2.5, where knowledge sources can be enumerated according to their characteristics and divided into six definite blocks:

1. The block of *passive* knowledge sources consists of repositories of information. This information might be extracted, for example, from the long-term observation of bottom-up processes or from applying musical knowledge to a current observation. Their contents are generated, modified and used by active (processing) knowledge sources.

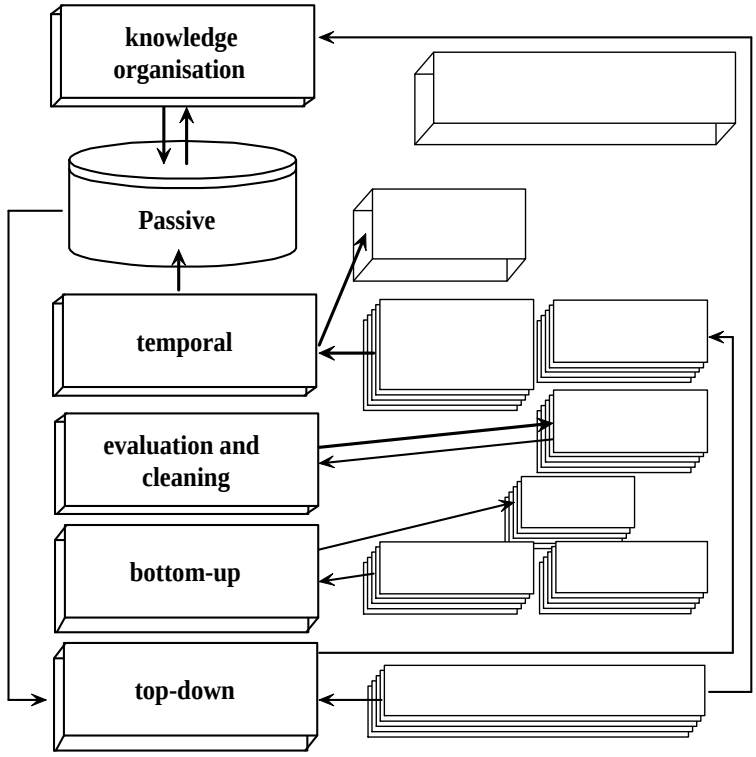

Figure 2.5: Blocks of knowledge sources and their interaction with the database.

The information in this block represents the listener's knowledge from experience.

Although these knowledge sources do not participate actively in the developing of a solution, they are responsible for the existence of expectations within the system. These expectations add flexibility into the system's "reasoning" hence increasing robustness. They are slowly built, and their information is never conclusive. It always implies a certain degree of uncertainty as it is based on assumptions itself.

2. The *bottom-up* processing block contains modules that fill empty levels of the hierarchy from the processing of information on the preceding data blocks. They work on "factual" information, sometimes taking data from the lower levels and processing it to generate a higher level

(e.g. obtaining peaks from an STFT magnitude profile), or from observing "evaluated" hypotheses on a mid-level and transferring them to a higher, more competitive, one (e.g. winning group hypotheses to note hypotheses level).

Most knowledge sources belong to this block. Robustness of these processing modules is necessary for accurate feature estimation at any level, as even the generation of experiencing knowledge depends on them. Bottom-up processing is the backbone of this system.

3. The *top-down* processing block contains modules that generate intermediate or high level data by combining information from lower levels in the hierarchy with knowledge stored in passive knowledge sources. Another function of these modules is to gather support for competitive hypotheses from lower levels of the hierarchy. Top-down modules are responsible for the non-linear thinking of the system.

   It is worth noting that the top-down function in this system is that of linking. Linking between high-level information (i.e. experiencing knowledge, group hypothesis) and low (factual) data in the framework. The idea is to generate highly competitive hypotheses, well supported by previous or sensorial information. However their output is never definitive and it needs to be evaluated in accordance with the rules of the competition.

4. The *evaluation and cleaning* block uses stored and self-contained knowledge to evaluate hypotheses at the different levels of the database. Based on its own evaluations, it deactivates "wrong" hypotheses. This block is in charge of the competition within the system. Competition is based on strong support and in the presence of explanatory fellow hypotheses: those that "explain" an evaluated hypothesis with their sole presence (i.e. harmonic interval, close strong peaks, etc).

   The rules of evaluation and cleaning are predefined, although they respond to contextual variables (i.e. the number of competitive hy-

potheses, the maximum energy of the group). They are associated with physical knowledge (that about the behaviour of audio signals), musical knowledge (that about the behaviour of music) or heuristics (the result of continuous experimentation or the programmer's experience).

5. The *temporal* processing block analyses information in the database across time. It provides data, both, for higher levels of the hierarchy and for passive knowledge sources. This is done by observing the data in short and long windows respectively.

   There are a number of reasons that justify the presence of temporal processing. It corresponds to the temporal nature of musical signals. It provides the context in which an observation makes more or less sense. It also allows a more general view of the development of the solution. The idea may seem obvious, but many signal processing methods are based solely on the frame by frame analysis of the signal, without consideration of its temporal variations. We do not regard this as adequate for our task.

6. The block for *knowledge organisation* is in charge of re-arranging passive knowledge sources depending on the context. It can generate temporal modified knowledge bases to analyse distinct sections of the input data. It also determines the minimal set of observations from experience that best help describe our process (i.e. from a large group of estimated notes, determining the set that describes a single instrument).

2.3.3 Scheduling

A scheduler is used as the control structure in this system. It is basically embedded within the main programming body of the algorithm. It constantly monitors the data within the hierarchy, firing the execution of the knowledge sources' action when convenient.

It aims to produce consistent and "certain" information at note hypotheses level for the whole length of the analysed signal. Consistency is evaluated as a function of invariability across time. The system applies the logic of the German proverb "Einmal ist keinmal": *what happens but once, might as well not have happened at all* [Kun84]. On the other hand, certainty is a function of resources being used: the implementation of both bottom-up and top-down processing to produce and confirm a certain hypothesis.

The scheduler sets action paths, where the most immediate action is determined according to priorities and possibilities. Priorities are associated with levels of the hierarchy. Possibilities associated with the data in those levels. The combination of both aspects determines the next step. Within this context, actions that resolve uncertainty associated with high-levels of the hierarchy are high in priority, while actions that involve the data currently available on the database are high in possibility. The system will execute the most high-priority task that is still possible.

As an example, one of the attributes needed to achieve our main goal is onset information. However, onset information cannot be generated without an onset detection function, which in turn is dependent on phase information, which is generated from the frame-by-frame analysis of the time-domain waveform. Here, an action route is generated: to select peaks in an onset detection function, to generate an onset detection function, to analyse phases on every frame of the Fourier analysis, to Fourier analyse the signal for a frame, to segment the signal waveform in frames, etc. As we move from top to bottom of the hierarchy, priorities get lower, but possibilities increase, as we reach actions that we can actually execute given the current status of the system.

In our simple example the decision-making process is straightforward, as there is only one possible route as depicted in Fig. 2.6. Thus, the actions will be executed from bottom to top, until onsets are generated.

However the type of reasoning needed to achieve "transcription" is often of a more complex nature. As mentioned, note hypotheses have different

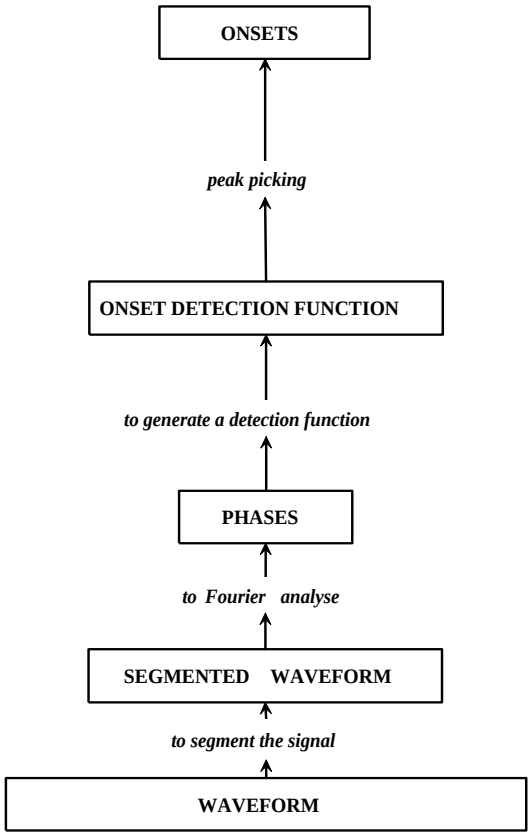

Figure 2.6: Scheduler example: a simple action path to onset detection.

attributes generated from a diverse range of operations. Let us extend our example to the calculation of the pitch of a note hypothesis. Pitch is only assumed to be true if it is consistently observed across time (*Einmal ist keinmal*). These individual observations are frame note hypotheses within our hierarchy. One of many possibilities is that such hypotheses are generated from the use of top-down processing (i.e. by using a note database within a passive knowledge source). But this knowledge is obtained from the long-term observation of the signal (a high number of frame observations), which, to put it simply, is obtained from the frame-by-frame analysis of the time-domain waveform. This generates a very complex list of actions that at the very bottom of the hierarchy are equal to the actions of the onset detection process. Therefore, segmenting the signal is not only contribut-

ing to generate onsets, but also to generate top-down pitch hypotheses (see Fig. 2.7), and in this regard is contributing to a task which is higher in priority. Whilst equally possible, actions related to high-priority tasks are executed before actions related to lower-priority tasks.

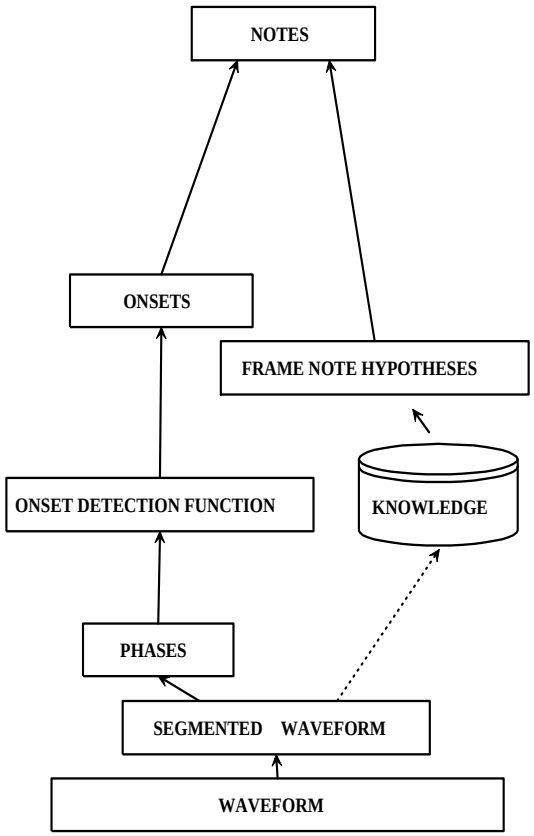

Figure 2.7: Complex scheduler example: Two action paths producing information necessary for note hypotheses. They both converge at low level.

This somehow glossy example is not only used to illustrate the complexity of the reasoning of the scheduler, but also to demonstrate, that no matter how high-level the required information is, its estimation can always be reduced to a set of very simple and straightforward actions, usually common to a number of processes within the system, supporting the idea of "parallel" processing within the blackboard architecture.

The scheduler is dynamically revising its action paths, hence deciding

whose action to start at every processing step according to the current status of the system. To attach some temporal logic to it, its tendency will be to move forward along the signal's length, keeping temporal windows of information in which to examine the status of the system. As frame information is more numerous, and largely irrelevant for the overall analysis, it is only kept within a short-term window. Note hypotheses and onset information is indefinitely kept until the main goal is finally achieved, when note information is transformed into an audible format.

This forward-moving pattern is only broken when, in its attempt to bring certainty into the equation, the scheduler decides to go back and forth along the signal's length (e.g. to generate top-down knowledge). This non-linearity is an asset of the blackboard architecture, moderately exploited in the current implementation.

2.4 Summary

In this chapter we initially explained comprehension as a hierarchical process: where the bottom is related to sensory information (from physical stimuli) and the top is associated with the assignment of meaning. Then, we introduced the two predominant approaches regarding the achievement of comprehension: the bottom-up approach that traces a linear path from the senses towards meaning, and the top-down approach that views comprehension as the cross-product between sensorial information and previous knowledge. We discussed strengths and weaknesses of both models, and adhered to the tendency that sees comprehension as the complex interaction between the two. This tendency is known as the interactive-approach.

The blackboard framework was introduced as an architecture that allows the automatic realisation of the interactive processing approach and applies it to the computational analysis of perceptual problems. Its main components: blackboard database, knowledge sources and scheduler were presented and discussed. We focused our discussion on auditory related tasks and briefly reviewed previous systems that used this architecture in order

to solve these. Ideas related to the author's early proposals of blackboard systems for automatic music transcription were introduced and discussed.

A more comprehensive framework based on these early attempts was then explained. The initial part of the framework is a two-dimensional database that considers information both in the time and frequency domain. It is hierarchically organised according to the complexity and the closeness of its information to an overall solution. The information summarises the observations and expectations of the system about the input signal. The second part of the framework consists of blocks of knowledge sources with both active and passive roles. They are responsible for the processing of the data and for the construction and storage of knowledge during the system's operation. Their detailed implementation will be the subject of subsequent chapters. Finally, the control structure is presented. It aims to achieve consistency and certainty for note hypotheses during the whole duration of the signal. Control is exercised by defining action paths directed towards the top of the hierarchy. The action most immediately executed is the one that is higher in priority while still possible according to the data present on the database. The creation of those action paths is dynamic and may include backward and forward temporal jumps. The architecture intends to successfully integrate the complex parts of the system as will be explained in chapters 3, 4 and 5.

Chapter 3

Time-frequency analysis

Feature estimation from music requires the analysis of the signal both in time and frequency. Any note identification algorithm depends, at some stage, on the spectral analysis of the musical signal (this is true even for our time-domain approach to be introduced in chapter 5). Therefore, the choice of an appropriate time-frequency representation determines the performance of the analysis system. An adequate front-end requires robustness, computational efficiency as well as proper management of the trade-off between frequency-domain resolution (which is high for pitch estimation of polyphonic signals) and time-domain resolution (necessary for an accurate detection of rhythmic features).

First, we will briefly review some of the proposed front-ends in the literature for the task of polyphonic music transcription, then we will explain the basics of our selected algorithm and justify our choice. Finally, we will incorporate the selected front-end to the framework described in chapter 2.

3.1 Previous approaches

3.1.1 Extensions of the Fourier transform

The analysis of signals with the Fourier transform and its extensions is the most classical and common approach in computer music analysis. The short

time Fourier transform (STFT), in Fig. 3.1, is a time-varying extension of the static spectral analysis performed with the Fourier transform. If considering a time-domain signal $s(n)$, its STFT can be calculated as:

$$S(n,k) = \sum_{m=-\infty}^{\infty} s(m)w(n-m)e^{-j2\pi mk/N} \tag{3.1}$$

where $k = 0, 1, \ldots, N-1$ is the frequency bin index and $w(n)$ is a finite-length sliding window. This representation allows the visualisation of frequency information as a function of time by isolating short segments of the signal. The window choice is crucial in order to reduce spectral leakage. The length of the window (N) determines the time and the frequency resolution. The longer the window, the better the accuracy in the frequency-domain. However, this occurs at the expense of the time resolution. Also, as the window length increases, the assumption of the stationarity of the signal during the analysis segment becomes weaker. This trade-off with a fixed-length window is the most important disadvantage of all spectrogram-based approaches.

An interesting extension of the Fourier transform, the phase vocoder, is a related technique that expresses the STFT coefficients as functions of their magnitude and phase. The strength of the phase vocoder lies in its potential to return a well-defined representation of the signal in terms of the time variations of its instantaneous amplitude and frequency [Dol82]. The technique has been the subject of numerous improvements regarding its theory, applications and implementation. A more detailed explanation of its functioning will be given in section 3.2.

3.1.2 Multi-resolution approaches

Several techniques have been proposed that make use of diverse concepts related to our understanding of the human hearing process. Brown [Bro91] proposed the use of the constant-Q transform for the analysis of musical signals. Its variable resolution along the frequency axis emulates the logarithmic frequency behaviour of the human ear. This is better explained

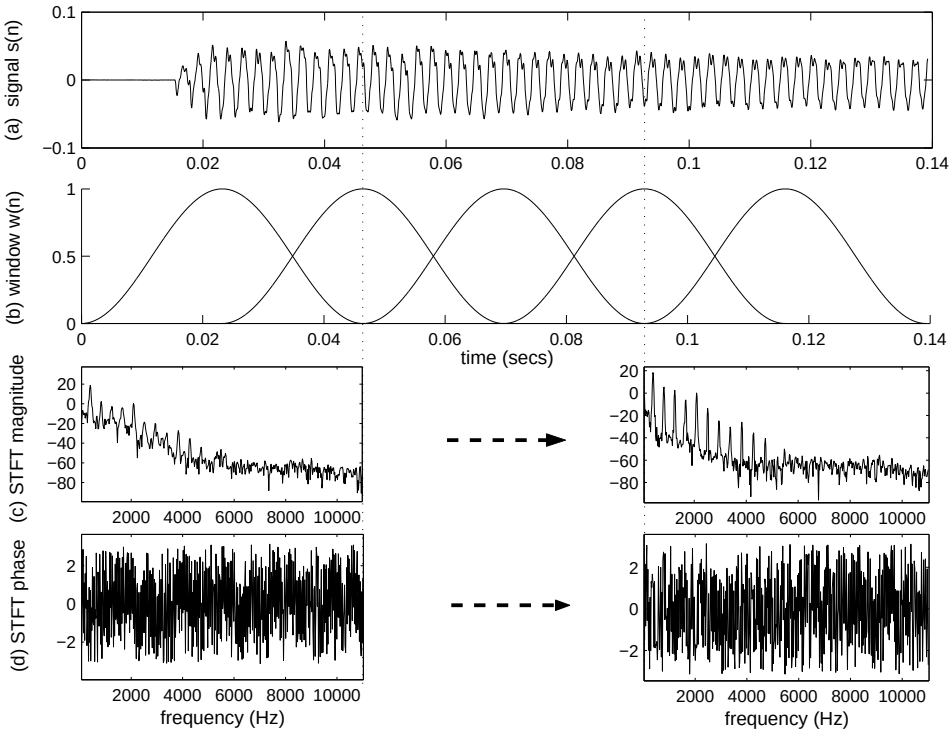

Figure 3.1: Short time Fourier transform of an audio signal (a). Overlapping windows (b) and the obtained magnitude (c) and phase (d) of two sample frames are shown.

in [Sta83] and [Sen85]. The frequency values correspond to the relation $f_k = (2^{1/24})^k \cdot f_{min}$, where f_{min} is the lowest frequency we want to resolve, and the maximum k is such that $f_{k_{max}}$ is less than the Nyquist frequency. The constant quality (Q) factor is defined as $f/\delta f$ where δf is the corresponding channel bandwidth. Therefore, the analysis window size becomes a function of k:

$$N(k) = \frac{f_s}{f_k} \cdot Q \tag{3.2}$$

where f_s is the sampling frequency. Our constant-Q transform is then defined as:

$$S_Q(n,k) = \frac{1}{N(k)} \sum_{m=-\infty}^{\infty} s(m) w_k(n-m) e^{-j2\pi Q m/N(k)} \tag{3.3}$$

Brown [Bro91] pointed out that to achieve accurate note identification, a quarter tone resolution is necessary ($Q = 34$). Efficient implementation of this transform is explained in [Bro93]. This technique provides a way of obtaining good frequency resolution without badly affecting our time resolution. However, it can be deduced from Eq.3.3 that this transform is not invertible [Bro91]. Therefore, its usability as an analysis-synthesis tool is limited.

An alternative constant-Q analysis approach is based on the use of the wavelet transform (WT). It has being used by several authors [KM88, Eva91] for the analysis and synthesis of musical signals, building upon the theoretical findings of Daubechies [Dau90]. It proposes that signals can be decomposed into dilations and translations of a generalised basis function or *mother* wavelet. Frequency is defined as a function of time (indexed by n) and scale (indexed by σ, and inversely related to frequency), such that:

$$f\left(\frac{n}{\sigma}\right) = \frac{1}{2\pi} w\left(\frac{n}{\sigma}\right) e^{-j2\pi k(\frac{n}{\sigma})/N} \tag{3.4}$$

Unfortunately this method, successful for the processing of musical signals, does not provide enough resolution to resolve high-frequency harmonics in polyphonic mixtures. This limits its applicability for the task of note identification.

Scaling is also used by Wilson et al [WCPD92, SW92] in the multiresolution Fourier transform (MFT). The scale expands the Fourier transform as:

$$S_\sigma(n,k,\sigma) = \sum_{m=-\infty}^{\infty} s(m) w_\sigma(n-m) e^{-j2\pi m k(\sigma)/N_\sigma} \tag{3.5}$$

where σ is the scale index or level, corresponding to the resolution of the representation. The scale modifies the length of the time response of the

window function $w_\sigma(n)$ (hence modifying the response in the frequency-domain). This time-frequency representation is used by Pearson [Pea91] and Keren et al [KZC98] for the transcription task. However, due to the multiple analysis levels, the amount of generated data is excessive, requiring high computational power and storage space for its processing.

3.1.3 Perceptual models

A step further into the recreation of the hearing process for the task of polyphonic music transcription is taken by Martin [Mar96b] (see Fig. 3.2). He modifies the log-lag correlogram originally proposed by Ellis [Ell96] as a tool for computer auditory scene analysis. It can be seen as a variation of the correlogram of Slaney [Sla93] and of the pitch perception model of Meddis and Hewitt [MH91]. Here, a gammatone filter-bank (of evenly-spaced filters in logarithmic frequency) is used to model the mechanics of the basilar membrane. It decomposes the signal into multiple frequency bands. The three axes of the correlogram volume are frequency (of the filter channel), lag (inverse pitch) and time. The inner hair-cell dynamics are modelled using a frequency/lag model. Their outputs are computed through the use of an envelope follower and a delay line of low-pass filters, whose outputs are, in turn, averaged to construct a "summary" autocorrelation. The system improves on regular correlograms by taking into account the pitch dependent ability of humans for note resolution. It tries to provide the means for the resolution of complex intervals in polyphonic mixtures by simulating perceptual features. This reduces the need for instrument models in the recognition part of a transcription system. Although the system is more robust against harmonicity problems (to be explained in chapter 5) than the usual sinusoidal analysis, at times it seems to trade harmonic overlapping problems with sub-harmonic overlapping problems (due to the use of lag instead of pitch).

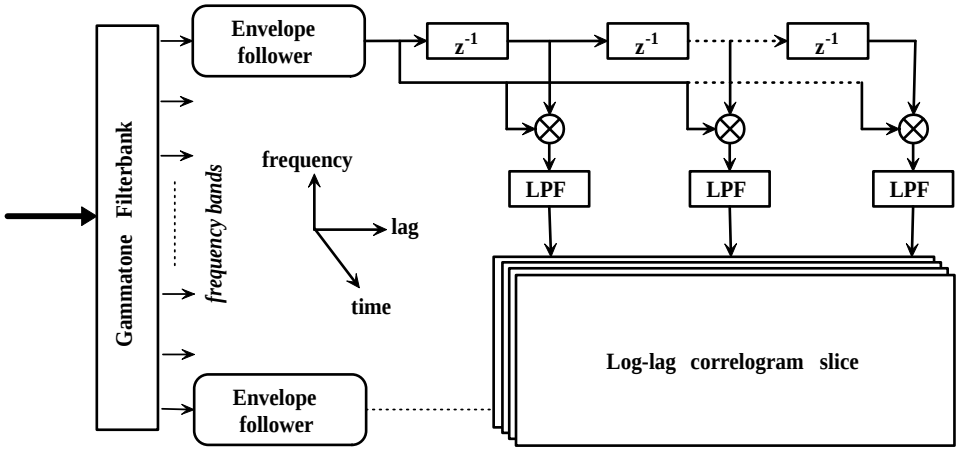

Figure 3.2: Block diagram of Martin's log-lag correlogram.

3.1.4 Bi-linear time-frequency distributions

The above-mentioned approaches are based on the averaging of the signal information over a finite-length window in time. This does not best represent the behaviour of a time-varying signal, which is not always stationary. Ville [Vil48] developed a deterministic time-frequency distribution in which the signal $s(t)$ is considered a complex pair of time-varying amplitude and phase $s(t) = a(t)e^{\varphi(t)}$ [Gab46]. The representation is known as the Wigner-Ville distribution and is defined as:

$$W_s(t,f) = \frac{1}{2\pi}\int_{-\infty}^{\infty} s\left(t+\frac{\tau}{2}\right) s^*\left(t-\frac{\tau}{2}\right) e^{-j2\pi f\tau} d\tau \tag{3.6}$$

The Wigner-Ville distribution (commonly referred to as the Wigner distribution - WD -) satisfies both its time and frequency marginals [PWS96]. This condition, not satisfied for the STFT, means that instantaneous power and power spectral density can be obtained from the distribution by integrating out the frequency or time variable respectively. Claasen and Mecklenbrauker [CM80] demonstrated that, at least for monophonic components, the instantaneous frequencies of the signal can be obtained by averaging frequencies of the WD. This provides better resolution than the STFT, which is theoretically just a smoothed version of this generalised representation

[PG99]. However, gains in resolution are not penalty-free. Due to the nonlinearities in the energy computations for this distribution (reflected in its bilinear character), interference terms appear in the distribution profile, sometimes causing negative values that affect the validity of the distribution as a physical representation of local energy [LPA93]. These cross-terms can appear at places where there is no spectral content. Obviously, these interference terms are highly undesirable for the task of note identification. Pielemeier and Wakefield [PW96] propose the so-called modal kernel, which intends to smooth the unwanted terms, without sacrificing the inherent advantages of the WD. The resulting representation is known as the modal time-frequency distribution (MTFD). It differs from other proposals [JW92, ZAM90, LPA91, LPA92, FR90, OW99] in that it is optimised for the estimation of instantaneous frequencies and amplitudes, regardless of the properties of the distribution (marginals, etc). In the MTFD, the effect of cross-terms is attenuated by smoothing in time, at the expense of the representation's time resolution. Sterian and Wakefield [SW97] improve the previous case by using a frequency-dependent smoothing kernel.

In all cases, it is assumed that the signal is well modelled as a sum of sinusoids [PW96]. Ideally, the smoothing algorithm should dynamically change depending on the signal's instantaneous conditions and not on a predetermined trade-off. However, implementation of such a kernel will require the use of complex models (i.e. statistical models), hence increasing the computational expense of the algorithm.

3.1.5 Parametric techniques

In these techniques, the input signal to our spectrum analyser is assumed to be the shaped power spectral density (PSD) of a white noise sequence. Shaping of the flat PSD is performed by a filtering operation as shown in Fig. 3.3. The observed signal is defined as a function of previous observations and filter coefficients. Marple [Mar89] approximated this process by using a rational transfer function model, that relates data by the linear difference

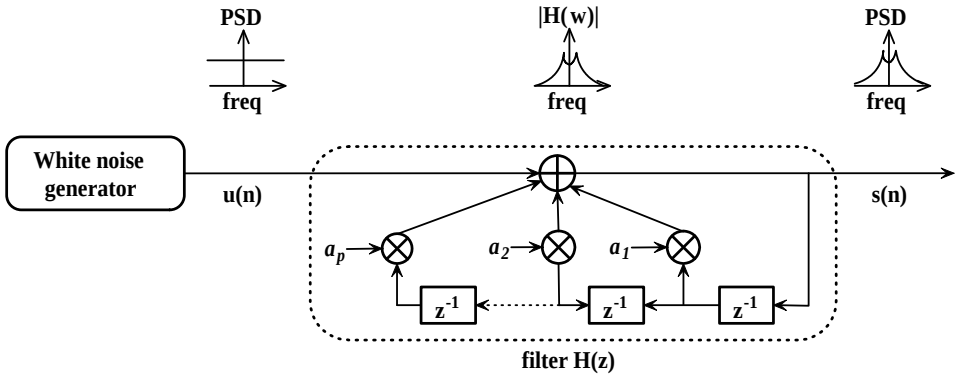

Figure 3.3: Auto-regressive model.

equation:

$$s(n) = \sum_{l=0}^{q} b_l u(n-l) - \sum_{k=0}^{p} a_k s(n-k) \tag{3.7}$$

where $s(n)$ is the observed sequence and $u(n)$ is the driving noise sequence. The model is known as the autoregressive moving-average (ARMA) model. It is used by Godsill [God97] for speech and audio signals. Its system function is defined as:

$$H(z) = \frac{B(z)}{A(z)} \tag{3.8}$$

where $A(z)$, the autoregressive branch, and $B(z)$, the moving-average branch are defined respectively as:

$$A(z) = \sum_{m=0}^{p} a_m z^{-m} \tag{3.9}$$

$$B(z) = \sum_{m=0}^{q} b_m z^{-m} \tag{3.10}$$

If all b_m, but $b_0 = 1$, are zero, then the process is called an autoregressive (AR) estimation of order p. This is the most used technique for parametric analysis. AR spectral analysis is used by Choi [Cho97] for fundamental frequency estimation in real time. Von Schroeter also applies it to music analysis [vS00]. He compares the performance of three AR parameter

estimation methods: the maximum entropy method, the MODCOVAR algorithm and Prony spectral line estimation [KM81, Mar87, Cyb84]; for the estimation of polyphonic spectra of real piano recordings, obtaining encouraging, although not extensive, results. The appeal of these methods rests on the fact that better frequency resolution can be obtained without considerable sacrifice to the time resolution. Furthermore, the resolution is not only a function of the length of the analysis segment, but also of the order of the model.

AR models represent an attractive choice for the music transcription task. However, these models have not being extensively used mainly because of the computational load introduced by some of the estimation algorithms when dealing with high-order filters, and due to the misrepresentations of non-harmonic components (such as those present during transients) that might be relevant for the analysis of complex musical signals.

3.2 The phase vocoder

For our implementation, we have chosen the phase vocoder as our time-frequency analysis tool. It consists of a framework for analysis and synthesis of audio signals. It was introduced by Flanagan and Golden [FG66] as a voice coding technique intended for speech processing. However, over the last decade, it has been widely used for the processing of music signals. The phase vocoder theory and its applications are extensively explained in the literature [AKZ02, Dol86, Cro80, CR83, PB98]. Here we only describe the basics of its functioning.

We can represent the complex spectral coefficient $S(n,k)$ in its polar form as $|S(n,k)| \cdot e^{j\varphi(n,k)}$. By studying the magnitude and phase of the time-varying sinusoidal components of $S(n,k)$, we can obtain more accurate estimates of their instant frequencies than those obtained using the Fourier magnitude spectrum alone. Furthermore, by individually modifying those features we can obtain useful variations of the original signal.

There are two different viewpoints that explain the phase-vocoder func-

tioning, they are known as the filter-bank and the Fourier-transform interpretation [Dol86]. They are introduced in the following sections.

3.2.1 The filter-bank interpretation

In this interpretation, depicted in Fig. 3.4, the phase vocoder is visualised as a fixed bank of N bandpass filters $w_k(n)$. The emphasis is on the temporal changes in magnitude and phase that occur on each of the singular filter bands. For the analysis stage, the input responses of the filters are defined as:

$$w_k(n) = w(n)e^{j2\pi nk/N},\ k = 0, 1, \ldots, N-1 \tag{3.11}$$

Each band's output $y_k(n)$ is obtained by filtering the input signal $s(n)$ with the corresponding filter:

$$y_k(n) = \widetilde{S}(n,k) = |S(n,k)| \cdot e^{j\tilde{\varphi}(n,k)} \tag{3.12}$$

where $\widetilde{S}(n,k)$ is the k^{th} complex-valued bandpass output sequence. The filtering operation is performed for all k by the convolution:

$$y_k(n) = \sum_{m=-\infty}^{\infty} s(m)w_k(n-m) = \sum_{m=-\infty}^{\infty} s(m)w(n-m)e^{j2\pi(n-m)k/N} \tag{3.13}$$

then, conveniently grouping the terms of the equation, we obtain:

$$y_k(n) = e^{j2\pi nk/N} \sum_{m=-\infty}^{\infty} s(m)w(n-m)e^{-j2\pi mk/N} = e^{j2\pi nk/N}S(n,k) \tag{3.14}$$

from where it can be observed that the phase of our bandpass sequence is given by:

$$\tilde{\varphi}(n,k) = \frac{2\pi kn}{N} + \varphi(n,k), \qquad \forall\, k \tag{3.15}$$

The analysis process can be implemented by heterodyning $s(n)$ with $e^{-j2\pi nk/N}$ followed by low-pass filtering on each channel. Finally, synthesis is simply performed by summing all the bandpass outputs:

$$y(n) = \sum_{k=0}^{N-1} y_k(n) = \sum_{k=0}^{N-1} S(n,k) e^{j2\pi nk/N} \tag{3.16}$$

For a real $s(n)$, the output signal can be yielded by adding the real-valued output signals $\hat{y}_k(n)$ over the range $k = 0, 1, \ldots, N/2$. This is demonstrated by Arfib et al [AKZ02].

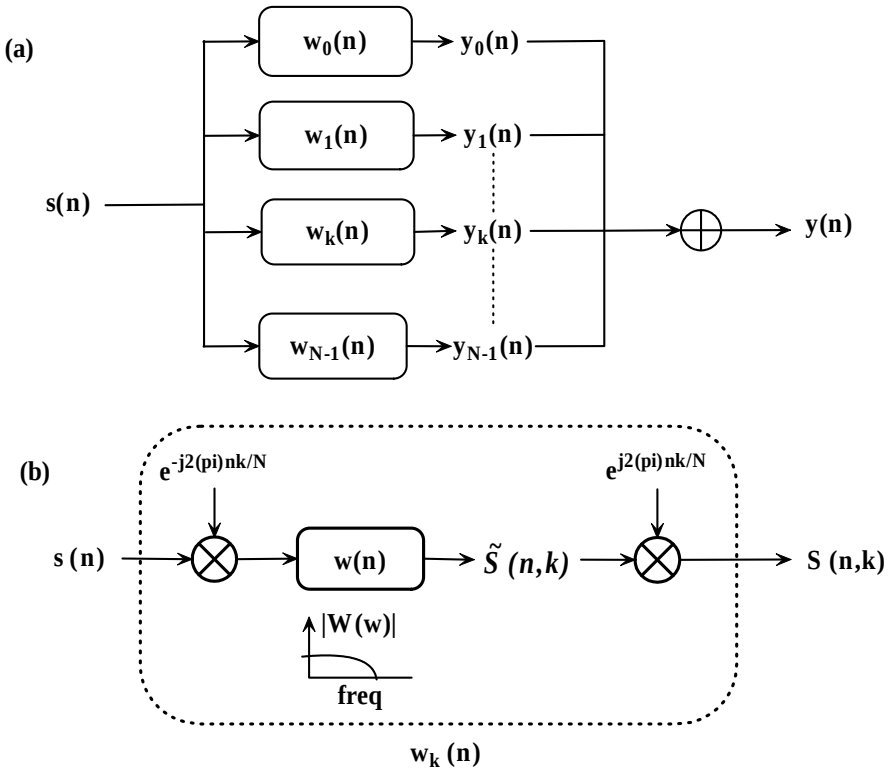

Figure 3.4: Phase Vocoder: Filter-bank interpretation

3.2.2 The Fourier transform interpretation

The second viewpoint, considers the phase-vocoder as a sequence of overlapping Fourier transforms taken over finite-length time windows (Fig. 3.5). Here, the emphasis is on the magnitude and phase values for all frequency channels at a particular time. The analysis part can be better understood by making some equivalences with the filter-bank interpretation. The time

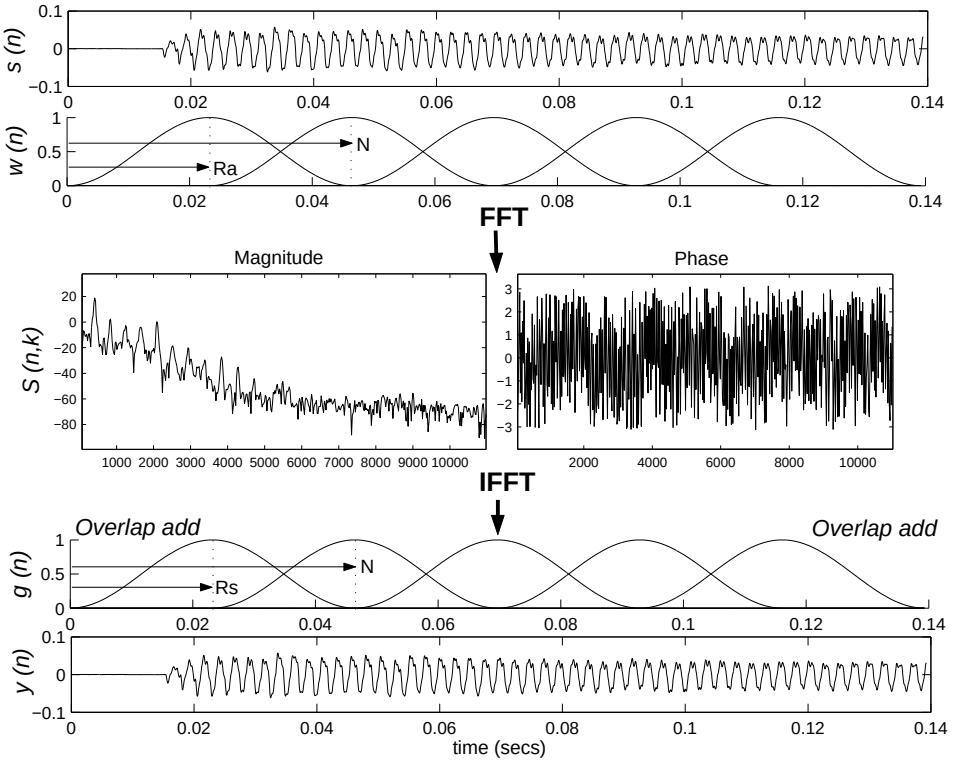

Figure 3.5: Phase Vocoder: FFT interpretation.

index n is now defined as τR_a, where R_a is the analysis hop size, and τ is the time index of the decimated STFT. Then it can be said that [Cro80]:

$$S(\tau R_a, k) = \sum_{m=-\infty}^{\infty} s(m)w(\tau R_a - m)e^{-j2\pi mk/N} = |S(\tau R_a, k)| \cdot e^{j\varphi(\tau R_a, k)} \tag{3.17}$$

where $k = 0, 1, \ldots, N-1$. The N points in the Fourier transform are equivalent to the number of evenly-spaced frequency bands of the filter-bank. The shape of the bandpass filters $w_k(n)$ depends on the shape of the windowing function applied to the $s(n)$ analysis segments.

$Y(\tau R_s, k)$ is obtained by modifying (processing) $S(\tau R_a, k)$. Let R_s be the synthesis hop size. Then, by applying the Inverse Fourier transform we obtain:

$$y_s(n) = \frac{1}{N} \sum_{\tau=-\infty}^{\infty} [Y(\tau R_s, k) e^{j2\pi\tau R_s/N}] e^{j2\pi nk/N} \tag{3.18}$$

These short segments are then weighted by a synthesis window $g(n)$ and overlap-added. The final output is calculated as:

$$y(n) = \sum_{\tau=-\infty}^{\infty} g(n - \tau R_s) y_s(n - \tau R_s) \tag{3.19}$$

The detailed analysis-synthesis procedure is explained in [Cro80] and [CR83]. By adopting the Fourier interpretation, we are allowed to implement the phase vocoder using the Fast Fourier Transform (FFT) algorithm (as proposed by Portnoff [Por76]), thus leading to a considerable increase in the computational efficiency. This is especially advantageous for large numbers of filters, as is the case for the analysis of polyphonic signals.

3.2.3 Instant frequency estimation

Eq.3.15 is an expression for the unwrapping of $\varphi(n, k)$. By unwrapping we mean the process of transforming the cyclic phase into a linear function (by adding the cumulative phase variation given by $\frac{2\pi k}{N} n = \Omega_k n$, as seen in Fig. 3.6). Ω_k is the frequency of the k^{th} sinusoid of our FFT analysis. If this sinusoid is stable over two consecutive FFT frames (whose times are $(\tau - 1)R_a$ and τR_a), then we can define a target phase for the present frame as:

$$\tilde{\varphi}_t(\tau R_a, k) = \tilde{\varphi}((\tau - 1)R_a, k) + \Omega_k R_a \tag{3.20}$$

where $\tilde{\varphi}((\tau - 1)R_a, k)$ is the unwrapped FFT phase of the previous frame. A deviation phase at the current frame, can be calculated by subtracting the target phase from the FFT phase:

$$\tilde{\varphi}_d(\tau R_a, k) = \text{princarg}[\tilde{\varphi}(\tau R_a, k) - \tilde{\varphi}_t(\tau R_a, k)] \tag{3.21}$$

The *principal argument* (princarg) function [GBA00] maps a phase value into the range $[-\pi, \pi]$. Given the target and deviation phase, an expected

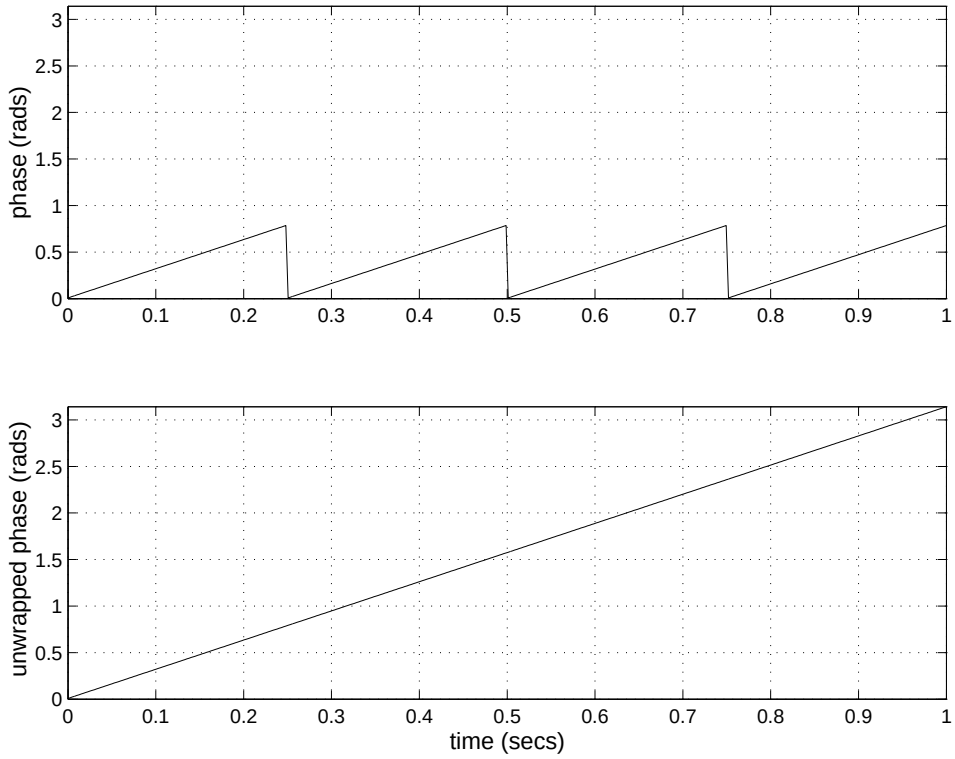

Figure 3.6: Phase Unwrapping.

unwrapped phase can be calculated as:

$$\tilde{\varphi}_u(\tau R_a, k) = \tilde{\varphi}_t(\tau R_a, k) + \tilde{\varphi}_d(\tau R_a, k)] \tag{3.22}$$

by substituting Eq. 3.20 and Eq. 3.21 in Eq. 3.22 and by subtracting the previous unwrapped phase $\tilde{\varphi}((\tau-1)R_a, k)$, the unwrapped phase difference between consecutive frames can be obtained as:

$$\Delta\varphi(\tau R_a, k) = \Omega_k R_a + \text{princarg}[\tilde{\varphi}(\tau R_a, k) - \tilde{\varphi}((\tau-1)R_a, k) - \Omega_k R_a] \tag{3.23}$$

Finally, we can define the instantaneous frequency as the rate of angular rotation, i.e. the unwrapped phase difference divided by the time between successive frames [Dol86]:

$$f_i(\tau R_a, k) = \frac{\Delta\varphi(\tau R_a, k)}{2\pi R_a} f_s \tag{3.24}$$

Theoretically, the error in the instantaneous frequency estimation is proportional to R_a/N, and never more than half the frequency resolution of the FFT analysis Δf [AKZ02].

3.3 Why the phase-vocoder?

Arguments can be made for all the reviewed time-frequency representations. Any choice will be a compromise as they all have different weaknesses and strengths. We will justify our choice as follows:

1. Frequency resolution: for polyphonic analysis, high-resolution is needed in the frequency-domain. In FFT-based representations, this can be obtained at the expense of time-resolution. This is the usual dilemma when choosing signal analysis tools. Constant-Q models (wavelets, constant-Q transform) pose an alternative to this problem. However, we consider that the task of note identification in polyphonic environments is better served by having this resolution evenly distributed along the frequency axis: in constant-Q analysis the ability to resolve in between two partials relies upon them being further apart at higher frequencies, which is not the case for polyphonic mixtures. We will favour frequency resolution over time resolution, as note identification is the core of our system. To cope with the time-resolution limitations of the phase-vocoder we implement an onset detection algorithm.

2. Computational load: we favoured a technique that can be easily and efficiently implemented using the Fast Fourier Transform (FFT) algorithm. The MFT and the log-lag correlogram are computationally expensive tools (very critical for the former). To a lesser extent the same can be said about high-order AR models and some of the kernels implemented for the smoothing of the modal TF distribution. This is very significant, as we will prefer to concentrate resources on the latter parts of the system.

3. Well-known technique: with all these interesting representations, the choice of the phase-vocoder as our time-frequency analysis tool may seem too conservative. Interestingly enough we will agree with that. While some of the techniques, such as AR models and Modal TFD are appealing (overcoming some of the disadvantages of FFT-based approaches), their applicability to musical signals is still in its infancy. WD Cross-terms for instance, are highly undesirable for the note identification task, and a satisfactory solution to this problem is arguably not yet proposed. Regarding AR spectral modelling, it can be argued that it implies a significant loss in spectral information that might be useful for the analysis of musical signals.

   The phase vocoder (PV) has been extensively studied and developed. In our implementation we make use of its analysis and synthesis properties (while some of the other techniques are only suited for analysis). PV applications (such as pitch-shifting algorithms) are used at different stages of our analysis process. Also, the phase information, the core of this vocoder definition, is exploited for the development of our own onset detection algorithm.

4. Note-identification strategy: representations such as the log-lag correlogram are intended to provide the transcription algorithm with information that closely resembles that used by our perceptual system. The idea being not to rely on instrument models or explicit music information to overcome the problems posed by polyphonic mixtures (notably harmonicity problems). However, we explicitly intend to rely on such models, in the form of knowledge about the analytic signal, to perform the automatic transcription task. We will invest our resources in the generation of this knowledge and in the understanding of the signal from a robust (although limited) representation. We do not intend to develop state-of-the-art TF representations (aiming to ease our task by effective pre-processing of the signal), especially when we do not consider that any of the above-mentioned tools manage to

do this without paying a significant price.

3.4 Integration into the blackboard framework

The time-frequency analysis of the signal provides the basic information that sustains the operation of our overall system. It equals the sensorial analysis that is performed at the lowest level of the human perception process. Onset detection and pitch estimation, as will be seen in the following chapters, entirely rely on the selected representation of the signal. In the blackboard architecture of the previous chapter, the information generated by the phase-vocoder analysis is related to the bottom levels of the hierarchy and obtained by the action of three bottom-up knowledge sources as illustrated in Fig. 3.7.

According to its current action path, the scheduler may select a new time position where it will perform the signal analysis. This information is sent to a segmentation knowledge source (**KS_segmentation**), that extracts information from the digitised audio file (in *wave* audio format) and places it in the *audio waveform* level of the database. Although simple, this knowledge source is not directly embedded in the phase-vocoder analysis, because its action will be required in different processes that do not necessarily involve Fourier transformations.

Once this data is on the blackboard, the scheduler is able to activate the **KS_PVFFT** knowledge source. This module calculates magnitudes and phases of the frequency-domain representation of the framed waveform. It uses the Fourier transform interpretation of the phase-vocoder, hence using the efficient fast Fourier transform algorithm (FFT) for its operation. It feeds from the *audio waveform* level and places its outputs into the *spectral phases* and *spectral magnitudes* levels.

The last knowledge source directly associated with the time-frequency analysis is referred as **KS_PVinstfreq**. As its name indicates, it uses the phases of two consecutive frames to calculate the bin-by-bin instantaneous frequencies for the current frame. Results are placed in the *instant frequencies* level.

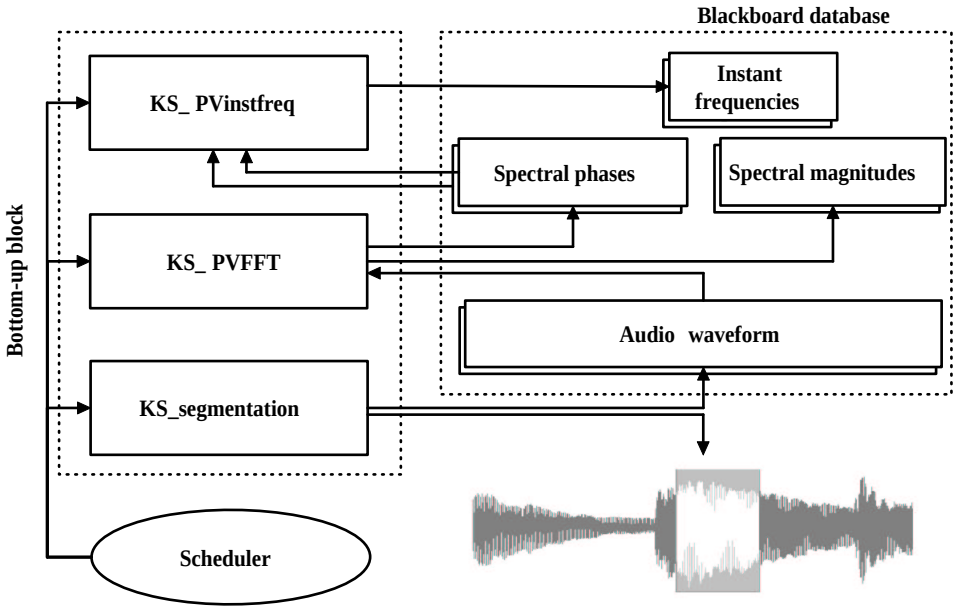

Figure 3.7: Integration of the time-frequency analysis into the blackboard framework.

3.5 Summary

In this chapter we started by describing the importance of time-frequency analysis for automatic feature estimation in musical signals. Robustness, efficiency and balance between time and frequency resolutions were mentioned as desirable for such a process. Different approaches for the time-frequency analysis of music were described and discussed. Their strengths and weaknesses emphasised.

The phase-vocoder, explained in its two interpretations: the filter-bank and the Fourier transform, was presented as our chosen front-end. It allows an accurate estimation of the features of the sinusoids involved in the Fourier analysis of the signal. The calculation of bin-by-bin instantaneous frequencies using the phase of consecutive frames is an example of this. This quality is advantageous for the estimation of more complicated features such as rhythm and pitch.

Our choice was justified by the following arguments: we deem resolution

at higher frequencies a necessity for multi-pitch estimation, hence an evenly spaced frequency representation is needed; for this stage of the system we want computational costs to stay low, freeing resources for subsequent and more complex processing, therefore an FFT-based approach is convenient; the phase-vocoder is a well-known technique and many tools have been developed from its theoretical basis, we intend to build upon this development in order to make the most of our process as will be shown in the following chapters.

Finally, it is shown how the phase-vocoder can be incorporated into our blackboard framework by summarising its operation in a few knowledge sources and by using its data to fill levels of the database.

Chapter 4

Note onset detection

The transcription task can be seen as a particular case of the auditory scene analysis problem. In order to elaborate a comprehensive representation of the current scene, it is necessary to gather as much information as possible from it. As specified in our definition of automatic music transcription (in chapter 1), events in our scene are defined by a number of features: onset time, pitch, energy and duration. This chapter, the first of two concentrating on feature extraction, will focus on the estimation of onset times. Its aim is to propose a novel and effective method for their detection.

It is structured as follows: first, the concepts of onset and transients are introduced; second, we review proposed systems, highlighting weaknesses and strengths; finally, we introduce our phase-based approach for onset detection and provide some examples to discuss its advantages and limitations.

4.1 About onsets and their detection

Providing an adequate and satisfactory explanation of what an onset means is an extremely difficult task. Let us start with a very simple definition: the onset is the precise moment when a new event begins. This immediately raises the need for a definition of *event* in this context. Mellinger [Mel91] defines an event as an auditory phenomenon that shows continuity for, at least, the smallest duration that can be perceived. Under this definition,

musical events may include expressive features (i.e. vibrato, legato, etc), timbre changes and notes. We are interested in identifying onsets related to the latter.

A note, such as the one depicted in Fig. 4.1(a), can be divided into its components: the transients plus noise component and the steady state. How these components model the sound is subject to discussion. One view regards all components to be present at any time. For this, *layered* case (Fig. 4.1(b)), individual sinusoids are independently analysed for the separation task [AKZ02, DDS01]. We will rather assume the *sequential* case, where the transients and the steady state are two separate, sequentially occurring, components of an event. This is shown in Fig. 4.1(c). Note that we are assuming the view where transients are integrated with the notion of attack. Attack transients are zones of short duration and fast variations of the signal spectral content (non-stationarity), where resonances are still building up [RJ01, Jeh97]. Their perception is caused by changes in the intensity, pitch or timbre of a sound [MR97]. Because of the unpredictability of such changes, they are difficult to model. Attack transients precede the steady state of the signal, when the signal is stationary, thus easily predictable. Sometimes, note onsets and attack transients are regarded as equal. However, we would like to emphasise that in our view, while these transients correspond to initial segments of notes, the onsets mark the beginning of these segments, thus marking the beginning of attacks themselves.

It is no wonder that, after facing so many difficulties in defining them, the task of detecting onsets becomes very hard to achieve. The boundaries between notes and different types of events are often ill-defined. The physics of the played instrument can introduce variations and modulations in a given sound without implying the presence of new notes. This can also occur as a consequence of the processing of the acoustic signal, recording conditions or just as an expressivity feature in the musical performance (i.e. vibratos in woodwind, brass and string instruments). Unsurprisingly, when dealing with polyphonic mixtures, the detection of onsets becomes increasingly difficult

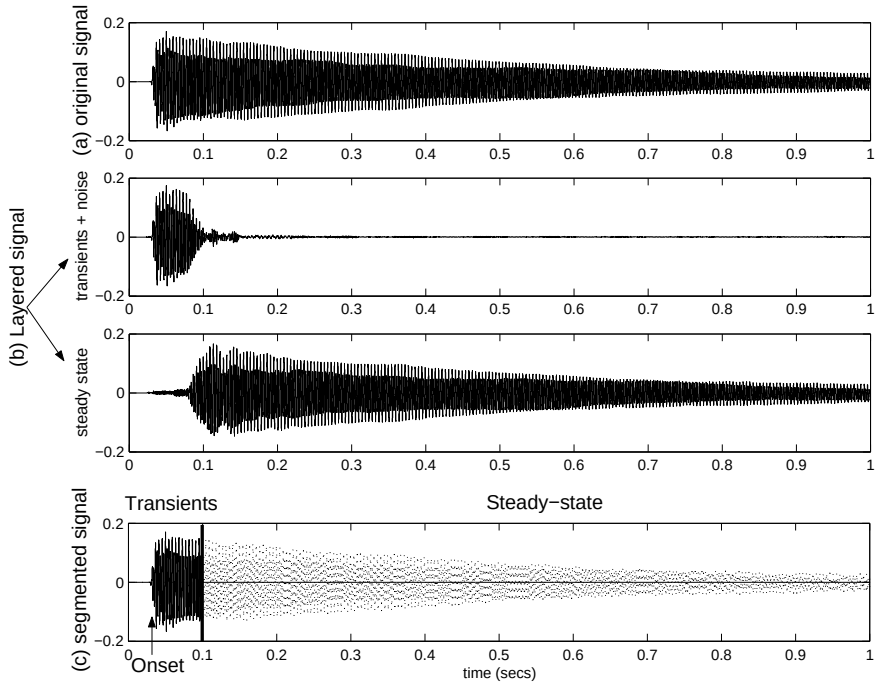

Figure 4.1: A musical note (a) and its components: the *layered* case (b) where the note presents steady-state and transients + noise components all along its duration; the *sequential* case where transients and steady-state are sequentially occurring events.

even for human listeners.

To provide a robust onset detection algorithm, careful attention has to be paid to the behaviour of attack transients. Some important observations were summarised by Masri [Mas96]. We will rephrase and extend them as follows:

1. Energy burst: in a note's energy profile, the highest concentration of energy can be found during the attack (when a steep increase can be observed). After that, energy progressively decreases. This can be observed in Figure 4.2(a). The more impulsive the components of the signal are (percussive sounds as opposed to tonal - more sinusoidal - sounds), the more sudden this increase-decrease energy characteristic

becomes.

2. Duration: the attack part of a note is usually very short, introducing significant changes to the signal (Fig. 4.2). This *abruptness* is a trademark of transients. It is particularly acute for percussive sounds.

3. Surprise: this is also related to the abruptness of transients, but from the statistical point of view. New events are unconnected to previous events, thus cannot be predicted from these. The proliferation of elements whose values are completely unexpected is more likely during transients.

4. Chaotic nature: during transients, the signal includes unstable chaotic elements, which quickly stabilise when entering the steady state (see Fig. 4.2(b)). These elements are not only highly uncorrelated with previous and future signal values, but also within different signal elements at a given time.

5. Steady-state: although obvious, an important characteristic of transients is that they are followed by the steady-state of the note. Chaotic components followed by chaotic components can account for noise, while a stable follow-up hints at the possible presence of a note.

How to capitalise on these observations has been the objective of previous researchers. We will provide a comprehensive summary of their research in the following section.

4.2 Review of different methods

4.2.1 Local energy

As mentioned before, the occurrence of an attack in audio signals is characterised by sudden changes in the signal's features, most noticeably an important energy increase. Hence, using the signal's energy is a straightforward method for attack transient detection. The local energy is calculated

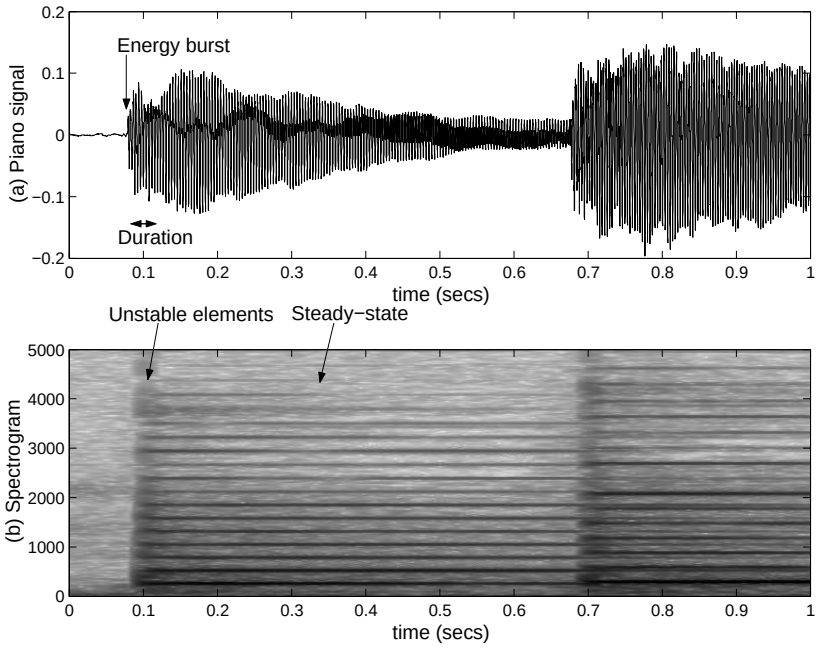

Figure 4.2: A sequence of two piano notes (a) and the corresponding spectrogram (b). The energy increase, short duration and instability related to transients can be observed as well as the stability of the steady-state part.

considering consecutive and overlapping segments of the signal, obtained by multiplying the input by a sliding window w_i (usually square). It is defined as:

$$E_n = \frac{1}{N} \sum_{i=-\frac{N}{2}}^{\frac{N}{2}} s_{i+n}^2 w_{i,n} \tag{4.1}$$

where n is the time at the centre of the window $w_{i,n}$, s_{i+n} is the ith sample in the current frame and N is the window size. The window smoothes the otherwise noisy and spiky profile of an audio signal. Unfortunately, this downgrades the one-sample accuracy by a factor of N. This is not necessarily a problem as the signal sampling intervals are usually much smaller than the ear's resolution between two onsets (2-10 ms) [Bre90].

Early onset detection systems [Sch85, MR97] used the energy profile for energy-burst localisation (the energy profile of a piano signal is shown in Fig. 4.3(b)). However, the observation of the signal's local energy does not provide enough reliable information for all the signals we might want to analyse. Also, peaks in the energy profile are better related to perceptual onset times than to actual physical onset times. It can be argued that perceptual onset times are not valid for processes that include some re-synthesis of the signal, as perception will be taken into account twice: in the analysis and in the hearing of the re-synthesised version of the sound. A way to improve this is by calculating the energy's first order derivative, to better extract noticeable variations in the signal's amplitude. The derivative is better related to physical onset times and is more accurate in tracking rapid increases in the energy profile. An example is shown in Fig. 4.3(c). A slight variation of this approach is proposed by Bilmes [Bil93], who uses a sliding window slope process to compute the slopes of the energy. His algorithm uses the linear least square method for the slope computation.

However, these approaches only seem reliable when dealing with percussive sounds (very fast energy increases), clearly separated in time and within very simple mixtures. Amplitude modulations (i.e. tremolo) or low-pitched notes are major causes of attack miss-detections. Also, when considering complex polyphonies, detection rates decrease strongly, even using slope-related values. An alternative is the simulation of the physics of auditory perception. According to Moore et al [MGB97], the perceived loudness change in the signal is approximately proportional to its intensity. An increase of 3dB (twice the signal's energy) corresponds to the perception of an event at twice the volume. The relative derivative of a function is equivalent to the derivative of the function's logarithm. Hence, calculating the difference of the signal's energy in dB roughly simulates the ear's perception of loudness [Kla99] (see Fig. 4.3(d)). Further improvement can be obtained by calculating the relative derivative of the energy: the energy difference divided by the current energy value. It highly reduces the effect of tremolo

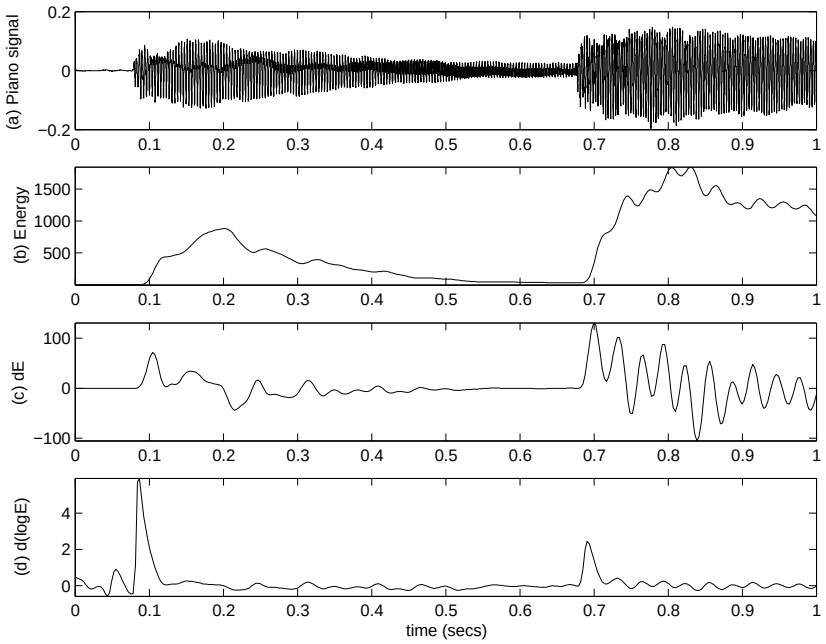

Figure 4.3: Piano signal (a), its energy profile (b), the first order derivative of its energy (c) and its relative derivative (d).

or amplitude modulation in the detection, as the overall derivative is attenuated. Sadly, even with these improvements, these simple methods are not effective when considering polyphonic, multi-timbral sounds and the acoustics of real live recordings (reverberation, distortion, etc). As a consequence, alternative, more complex methods are required to successfully identify onsets.

4.2.2 Analysis in the frequency domain

Recent onset detection methods have improved on the previous case by relying on the observation of the signal's energy behaviour in the frequency-domain. We will briefly review some of these ideas and discuss their robustness.

The STFT performs a frame-by-frame harmonic analysis where station-

arity is assumed. The introduction of a new sound element is unconnected to the preceding sounds, hence it cannot be predicted from them. The non-stationarity causes distortion in the FFT spectrum. An attack-related energy increase appears as a broadband noise all along the frequency axis. This attack transient *noise* is particularly noticeable at high frequency locations [RJ01], since at low frequencies, high concentrations of energy (in the bins corresponding to the first few harmonics of the played note) mask this effect. Masri [Mas96] proposes that these observations could be emphasised by weighting the energy values in the frequency-domain, as shown in equation 4.2:

$$\widetilde{E}_n = \frac{2}{N} \sum_{j=1}^{\frac{N}{2}} |S_{j,n}|^2 W_j \tag{4.2}$$

where the $S_{j,n}$ are the FFT coefficients of the windowed signal (centred at time n), and W_j is the frequency-domain weighting window. For $W_j = 1 \; \forall \, j$, this equation is equivalent to the energy function (by using Parseval's theorem). Masri suggests the use of $W_j = j \; \forall j$, therefore generating a *High Frequency Content* (HFC) function, that linearly emphasises the contribution of each frequency bin. If compared with energy, this HFC function has greater amplitude during the transient/attack time. The HFC of a test piano signal can be seen in Fig. 4.4(b). A detection function is then built as:

$$D_n = \frac{HFC_n}{HFC_{n-1}} \frac{HFC_n}{E_n} \tag{4.3}$$

It presents sharp spikes corresponding to sound attacks. These spikes can then be filtered according to a fixed threshold. The detection function presents a slightly noisy profile, requiring a certain amount of post-processing to perform successful peak picking (see Fig. 4.4(c)). It is important to notice that the robustness of this approach is compromised when the sound's own energy distribution masks the expected onset-related changes. This is the case for noisy recording environments and high-pitched sounds

(i.e. open cymbals in pop music).

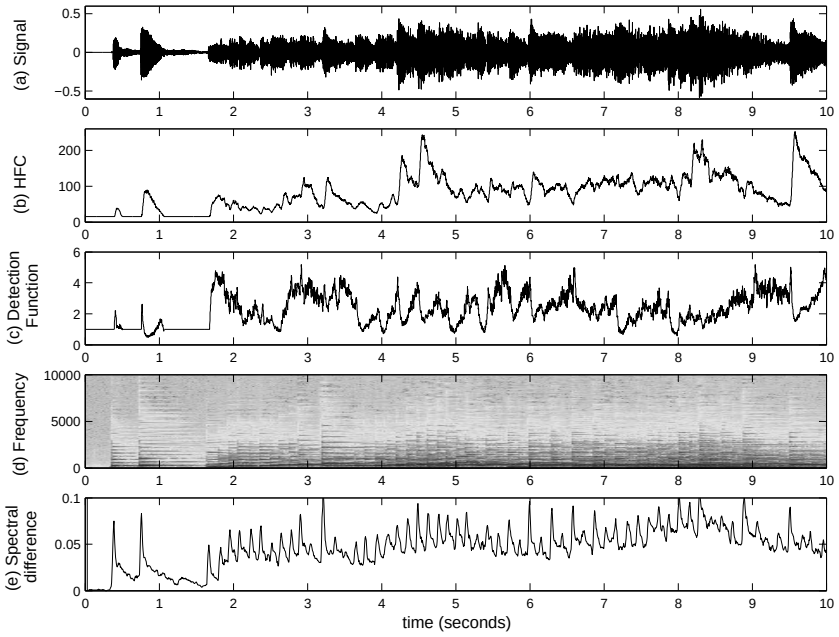

Figure 4.4: Piano signal (a), its high frequency content (b), the profile of Masri's detection function (c), the spectrogram of the signal (d) and its dissimilarity function (e).

An alternative view suggests that the observation of frame-by-frame spectral variations, regardless of their location in frequency, allows robust transient detection. The idea being to capitalise on changes in the spectral content as well as in sudden energy increases. By performing this two-fold analysis the probability of successful detection of attacks is increased. Masri adheres to this view in the same thesis where the HFC function is proposed. He proposes the construction of a dissimilarity function, as the relative difference between bins of the signal's spectrogram (in Fig. 4.4(d)). A threshold is used to filter the resulting waveform. This method is able to detect sharp energy changes and tonal changes when the spectral difference is noticeable enough. This is less reliable for harmonically related intervals. An example

of the dissimilarity function being applied to a musical signal is shown in Fig. 4.4(e).

Rodet and Jaillet [RJ01] propose a method where STFT frequency bands are independently analysed in time. During a short window around the time of the attack, a triangular shape is fitted to the energy profile of each frequency channel using the minimum least square method (Fig. 4.5). A detection function is built as:

$$I_{f,m} = \frac{(M - M_a)(M - M_b)}{M_a + M_b} \tag{4.4}$$

where M is the maximum peak, M_a the average amplitude before the attack and M_b the average amplitude after the attack. In this case the transient analysis is performed on each band and the results are aggregated across frequencies and within an *uncertainty* time interval. In their article it is mentioned that the high-frequency estimates were more reliable (reliability being proportional to the good-detection/false-alarm ratio). This is coherent with Masri's Energy-distribution approach (HFC).

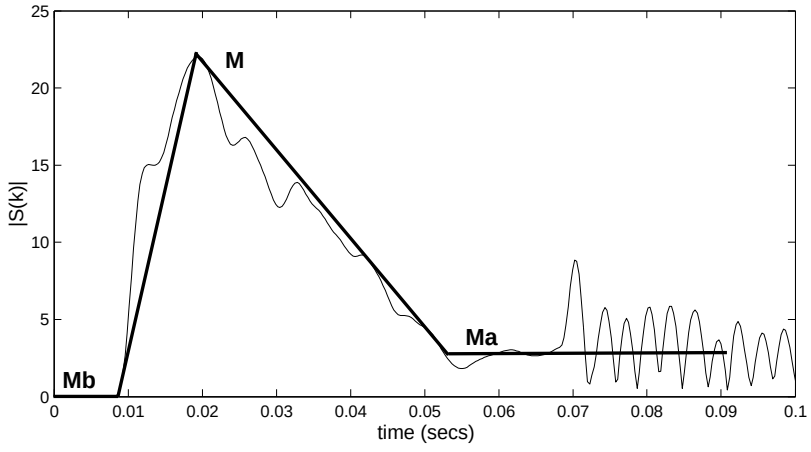

Figure 4.5: The magnitude of an STFT band $|S(k)|$ during an attack. A triangular shape is fitted with a maximum peak M and average amplitudes before (M_b) and after (M_a) the attack.

This convergence between Masri's and Rodet's observations requires

some attention. Clearly, the way the frequency-domain information is grouped (or how the observation is emphasised) greatly affects the reliability of the detection. At first sight, it seems logical that the observation of the behaviour of the signal at high frequencies returns better results. Disappointingly, Masri results favour the spectral dissimilarity approach over the use of the HFC function. From his examples, it is evident that the success of the detection is subject to the signal's distribution itself. As a consequence, some researchers have tried to tackle the problem using perceptually and biologically more meaningful systems, such as multiple-band filter-bank models.

Bilmes [Bil93] suggested an early multiple-band onset detection algorithm for drum signals. His system only considered two bands: for low and for high frequencies. This proved ineffective. Goto and Muraoka [GM95], sliced the spectrogram of a signal into *spectrum strips* and searched for sharp energy changes. The detected onsets were used in a multiple-agents architecture aiming to estimate rhythmic patterns. Scheirer [Sch98] implemented a six-band filter-bank (using sixth-order elliptic filters) and auditory-inspired processing to produce onset trains. The resulting data was feed into comb-filter resonators to estimate the tempo of the signal. Although one-by-one onset detection was not the goal of his research, Scheirer shaped future onset-detection systems by pointing out that independent analysis of perceptually meaningful separate frequency bands was necessary to successfully identify attack times (by recreating the human perception of onsets).

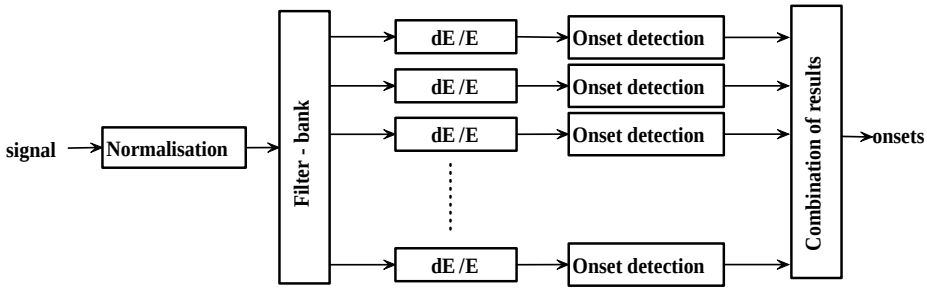

Figure 4.6: Block-diagram of Klapuri's perceptual onset detector

Klapuri [Kla99] considered the human hearing process in his implementation of a *perceptual* onset detector. In this system, the overall loudness is normalised using the model proposed by moore et al [MGB97]. A filterbank divides the signal into 21 non-overlapping bands. A relative difference function is used at each band and thresholded to yield times and intensities for transient components. Band-wise results are cleaned, according to masking principles, and finally combined. This process, perceptually-significant bands apart, is similar to the process performed by Rodet. The whole system is illustrated in Fig. 4.6.

4.2.3 Detection through signal modelling

In all the reviewed systems, analysis is performed over all components of the signal: steady state, transients and noise. However, some components carry information that is more relevant to this analysis than others. An alternative view for onset detection proposes that attack times can be easily detected by straight energy analysis methods when processing only the transient part of the signal. To obtain such information, researchers have attempted to model audio into its components. In this section, we will review methods concerned with signal modelling, and how they deal with the representation of transients.

The additive synthesis method [MQ86] proposes the modelling of signals by a number of sinusoidal oscillators. This approach is adopted by Serra [Ser89], who suggests the Spectral Model Synthesis (SMS) method, extending the original signal representation by adding noise. The resulting model can be described as:

$$s(n) = \sum_{k=1}^{N} a_k(n) cos(\varphi_k(n)) + r_k(n) \tag{4.5}$$

where a_k and φ_k are the slowly time-varying amplitudes and phases of the oscillators (corresponding to the stable components of the musical signal), and r_k is the noise or residual, obtained by subtracting the estimated steady state from the original signal.

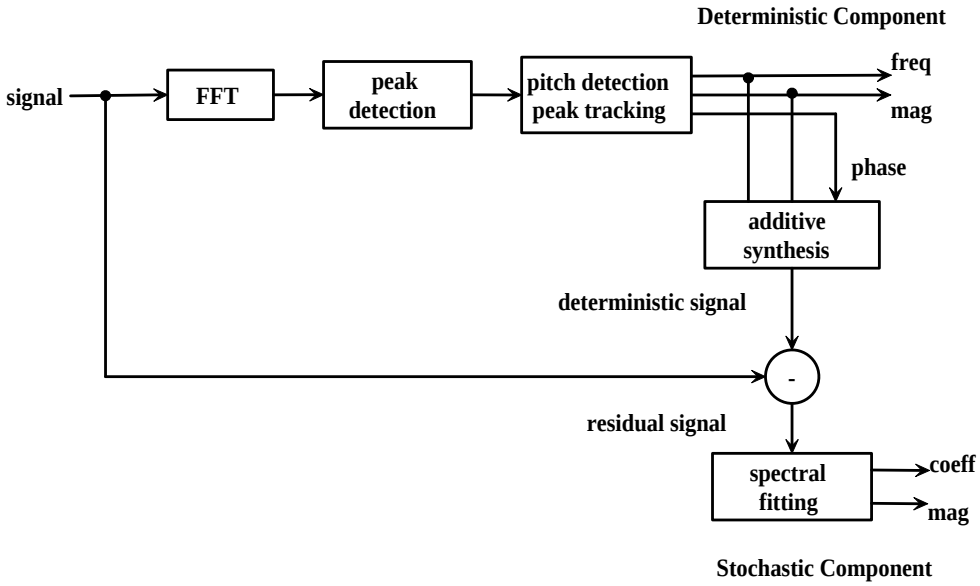

Figure 4.7: Block diagrams of Serra's sinusoidal analysis.

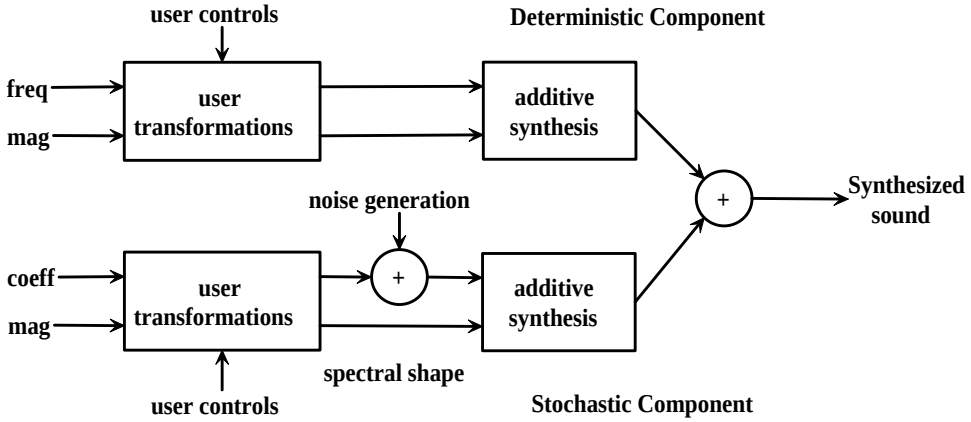

Figure 4.8: Block diagrams of Serra's sinusoidal synthesis.

The residual, a transient plus noise component, is modelled as slowly-varying filtered gaussian white noise. The block-diagram of the analysis and the synthesis implementation can be seen in Figures 4.7 and 4.8. Efficiency in the implementation can be achieved by using the Inverse Fourier Transform instead of a bank of oscillators [FRD92]. A variation is proposed by Levine [Lev98] where an octave-spaced filter-bank is used as part of the

system's front end. The resulting method is known as multiresolution sinusoidal modelling. He implements an onset detection system that evaluates the short time energy of the residual. The ratio between residual and original short-time energies tends to zero when original and synthesised signals are close (as expected during the steady state). A ratio near 1 implies that the signal was not well modelled with sinusoids, and that the corresponding frame (or group of frames) is likely to be a transient.

The quality of any additive-synthesis-based modelling depends on the estimation of the sinusoidal parameters. The standard algorithm detects peaks in the frequency-domain and uses them as evidence for the presence of sinusoids. The parameters of the corresponding sinusoids are estimated by quadratic approximation [MD92]. Finally peaks are grouped in time into *tracks*, determining the beginning and end of the sinusoids' *lives*. As the approach is based on frame-by-frame peak picking, peaks corresponding to transient information are also selected, even when they do not represent the stable components. This affects the model considerably as the length of the frame is usually long for steady-state analysis. To face this problem, the grouping of peaks can be improved by using Hidden Markov models [DGR93] or by implementing a sinusoidal likeness measure (SLM) to differentiate between peaks corresponding to sinusoids and impulses in the frequency domain [Rod97]. However, by definition additive synthesis is conceived for steady state modelling thus not allowing a comprehensive representation of transients. In order to achieve better modelling it seems that the use of an explicit model for transients is required.

Verma et al [VLM97] complemented the SMS approach by introducing the concept of Transient Modelling Synthesis (TMS). This method exploits the impulsiveness of transients in time, by performing sinusoidal modelling over the Discrete Cosine Transform of the residual. It then fits a parametric model to transients. The synthesised transients are subtracted from the residual, creating a *pure* noise residual. Hamdy et al [HAT96] also extended the sinusoidal analysis approach by using an explicit transient model on

the residual signal. It encodes the wavelet transform of the residual using edge prediction and noise modelling, decomposing it into transient and noise components. The used wavelet-packet tree has frequency bands matching the critical band structure of the human auditory system. Daudet [Dau01] proposes a similar approach where wavelet trees are used to detect transients from the residual. He suggests that transients are related to large coefficients at different scales of the wavelet representation of the residual. This can be seen in Fig. 4.9. By connecting those coefficients, structures are found that describe the signal's transients. Small, unconnected values are then eliminated from the tree structure by a pruning process.

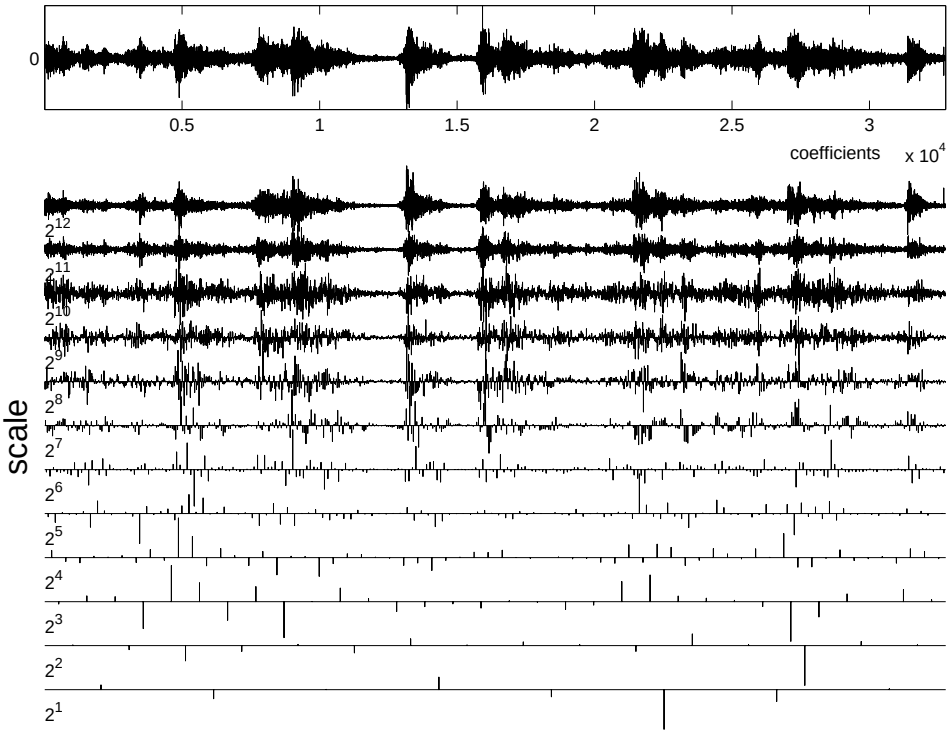

Figure 4.9: Set of wavelet coefficients. Original signal minus tonal part (top) and coefficients at scales 2 to 2^{12}. From [Dau01] reprinted with permission.

An alternative to the modelling of signals with sinusoids (and residual analysis) is through the use of matching pursuits (MP). Matching pursuits is an algorithm that decomposes signals into predefined grains of a large dic-

tionary [MZ93]. The grains or atoms that better approximate the analytic frame are iteratively selected (through correlation) and subtracted from the original signal. The process is repeated until the residual is minimised. Usually, dictionaries are made from scaled, modulated and translated versions of a single window function known as Gabor expansions. These data-sets comprise both Fourier and wavelets bases. However, the even-symmetry of the window causes the atoms to be symmetric in the time-domain. This is an undesirable feature when modelling asymmetric components such as transients. To compensate for this characteristic, Gribonval et al [GBM+96] propose the use of the High Resolution Matching Pursuit (HRMP) algorithm, that controls the *greed* of the standard MP method. This is done by implementing a correlation function that not only considers the amount of energy the match *takes* from the signal, but also the amount of energy it adds, favouring a better fit on each iteration over a rapid decrease in the residual's energy. Goodwin [Goo97, Goo96] tackles the symmetry problem by including asymmetric atoms into the dictionary. These atoms correspond to dampened sinusoids, that provide a better fit to the transient profile. With matching pursuits, the larger the dictionary, the better the chances of finding a successful match for the current residual. Predictably, the size of the dictionary affects the computational expense that this approach implies. Efficiency is the main problem for these algorithms.

4.2.4 Statistical approaches

Recent approaches have analysed the statistical behaviour of audio signals aiming to determine their underlying semantic structure. Within this context, the task of onset detection is reduced to the observation of abrupt changes in the statistical character of an otherwise consistent data-set.

Jehan [Jeh97] models the signal using a parametric technique and then suggests the use of segmentation algorithms that statistically analyse the model searching for pronounced parametric changes. One of his suggested segmentation methods, uses Brandt's algorithm [vB83]. It is illustrated in

Fig. 4.10. A fixed length segment is divided at time r. The two resulting segments are modelled using an autoregressive model plus gaussian white noise. The parameters are the autoregressive coefficients and the noise variances. The algorithm finds the distance r and the set of parameters that maximises the log-likelihood ratio between the probability of having an onset at point r and the probability of not having an onset at all (shown in Fig. 4.10(b)). Onsets are detected when the maximum likelihood surpasses a fixed threshold.

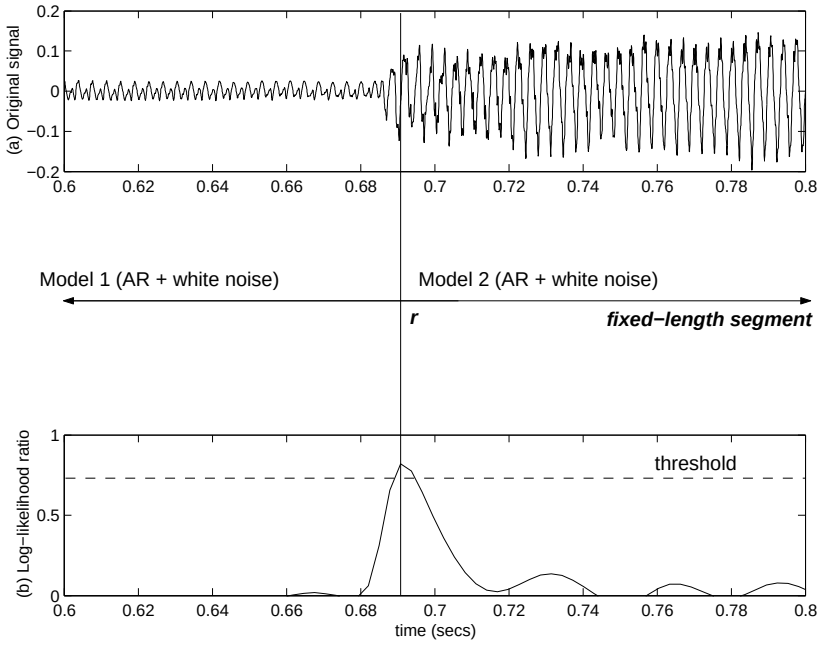

Figure 4.10: Segmentation method using Brandt's algorithm. The system finds the distance r and the two sets of parameters (Models 1 and 2) that maximise the log-likelihood ratio (b).

A second segmentation method is based on the divergence test. This approach is also explored by Thornburg and Gouyon [TG00]. An autoregressive model is developed that fits the data up to a certain point r. If a change is present at time r, the model will start misrepresenting the data set.

How well the data is represented is measured through the log-likelihood ratio between the current model and the initial model (that can be a standard gaussian model). A change of sign on the ratio shows a model mismatch. A new cumulative model is then started (and evolved until a new change occurs) and an onset proposed on the position r of the measured model change.

Abdallah and Plumbley [AP03] propose a method based on measuring the *surprise* introduced by onsets when observing a musical signal. In their system, the data is represented by using Independent Component Analysis (ICA), a method that linearly transforms the data into statistically independent components [Com94]. A double-sided exponential distribution is used as a statistical model that fits the data representation. It approximates a super-gaussian distribution that recreates the behaviour of audio signals. The model is described as:

$$P(x(k)) = \frac{1}{C_N} \prod_i e^{-|x_i(k)|^\alpha} \tag{4.6}$$

where $x_i(k)$ is the i^{th} observation of the k^{th} segment, α is an adjustable parameter, and C_N is a normalisation constant. The unexpectedness of the event is measured using the *surprisal* of the event function as proposed by Attneave [Att59]:

$$S(x(k)) = -\log P(x(k)|\{x(k-1), x(k-2), \cdots\}) \tag{4.7}$$

Initially, a memory-less model is considered where the history of the signal is ignored. In this case, only $P(x(k))$ is taken into account. If considering our super-gaussian model, then our surprise signal is described by:

$$S(x(k)) = \sum_i |x_i(k)|^\alpha + \log C_N \tag{4.8}$$

This function is expected to show sharp peaks when an unpredictable event occurs. Finally Abdallah and Plumbley propose that by using a conditional density model, that takes into account the recent history of the

signal, the sharpness of the *surprise* signal is increased, thus allowing better onset detection. An unsupervised method is used to detect clustering in the solution space.

4.3 Phase-based onset detection

Arguments can be made for all reviewed methods. Energy-based algorithms are usually fast and easy to implement. However, their effectiveness decreases when the transients of the analysed signal are not pronounced (i.e. non-percussive sounds) and when energy bursts of different events overlap in polyphonic mixtures. Model-based and statistical approaches offer conceptually strong and often more general methods for onset detection. However, in some cases they require high computational power or a dedicated framework (i.e. wavelets, matching pursuits, ICA) that is not necessarily easy to implement.

In this section we aim to propose an algorithm that builds upon the strengths of some of the above-mentioned methods. We intend to do this by:

1. Using phase information: attack transients are well-localised events in time. The phase carries all the timing information of an audio signal. When analysing in the frequency-domain this information is usually ignored. We suggest that phase analysis can return more meaningful results for the detection of new events than solely relying on energy values. Furthermore, as analysing phase implies a type of frequency analysis, changes that are not as noticeable as energy bursts can still be successfully detected as *pitch* bursts.

2. Building on a segmentation algorithm: we will develop our onset detection system upon a well-known method for phase-based transient / steady-state (TSS) separation [SL94]. The model assumes an expected behavioural pattern for the steady-state of the signal. When an attack occurs the pattern is disrupted. By using this technique, we

try to take advantage of the conceptual strength that signal modelling implies, hence enhancing the robustness of our approach.

3. Using an FFT-based framework: as explained in the previous chapter, we use the phase vocoder as our time-frequency representation. Our onset detection system is built into this framework. This allows us to pursue our goals with an easy-to-implement algorithm that allows FFT-based fast computations.

4. Using statistics: the data produced by the segmentation algorithm is analysed using simple statistical methods. By relying on the statistics of our data distribution we intend to generalise our analysis to a large variety of signals. In our view, this is a very useful way of summarising observations that help to describe the signal's behaviour.

In the following sections we will describe in detail the theory and observations behind the proposed onset detection approach.

4.3.1 TSS separation

To clarify a somehow deceptive title, it is important to emphasise that we do not perform transient / steady-state separation. We are just using the theory behind the TSS method proposed by Settle and Lippe [SL94] to generate a distribution function that suits our requirements. The method is described in detail by Arfib et al [AKZ02] and improved by Duxbury et al [DDS01] by using a multi-resolution front end (provided by a constant-Q filter-bank). In this section we will explain this theory as relevant to our implementation.

Let us return to the last chapter's phase vocoder calculations. According to the analysis presented there, the instantaneous frequency of the k^{th} bin can be expressed as:

$$f_i(\tau R_a, k) = \frac{\Delta\varphi(\tau R_a, k)}{2\pi R_a} f_s \tag{4.9}$$

where $\Delta\varphi$ is the unwrapped phase difference between two consecutive FFT frames. The FFT phases of such frames: $\tilde{\varphi}(\tau R_a, k)$ and $\tilde{\varphi}((\tau-1)R_a, k)$, can be used to calculate the unwrapped phase difference as follows:

$$\Delta\varphi(\tau R_a, k) = \Omega_k R_a + \text{princarg}[\tilde{\varphi}(\tau R_a, k) - \tilde{\varphi}((\tau-1)R_a, k) - \Omega_k R_a] \quad (4.10)$$

The *principal argument* function (princarg) maps onto the $[-\pi, \pi]$ range. Consider the behaviour of the individual k^{th} sinusoid of our phase vocoder representation. If the sinusoid is stable, it is naturally expected that the instantaneous frequency at time τR_a will be very close (if not equal) to the instantaneous frequency at time $(\tau-1)R_a$. Inversely, if the sinusoid is not stable (i.e. when a new, unpredictable event occurs), the variation between these two frequencies is expected to increase. An example of this can be seen in Fig. 4.11. It shows the instantaneous frequency of the bins corresponding to fundamental frequency and the first two partials of the played note. Note how the instantaneous frequency varies around the onset time. This can be expressed mathematically by defining the instantaneous frequency difference Δf_i between consecutive frames as:

$$\Delta f_i(\tau R_a, k) = f_i((\tau-1)R_a, k) - f_i(\tau R_a, k) \quad (4.11)$$

such that:

$$|\Delta f_i(\tau R_a, k)| \begin{cases} = 0 & \text{during steady-state} \\ > 0 & \text{during attack transients} \end{cases}$$

By combining the concepts of instantaneous frequency difference (Eq. 4.11) and unwrapped phase difference (Eq. 4.10), the following is obtained:

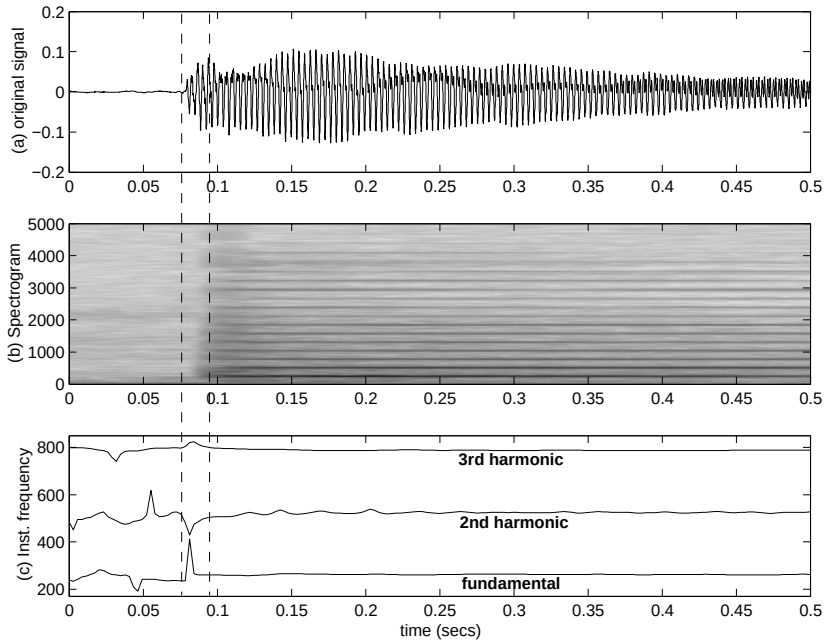

Figure 4.11: The instantaneous frequency (c) of the bins corresponding to the fundamental and first two partials of the note depicted in (a) and whose spectrogram is shown in (b).

$$\Delta f_i(\tau R_a, k)\frac{2\pi R_a}{f_s} = \Delta\varphi(\tau R_a, k) - \Delta\varphi((\tau-1)R_a, k) \quad (4.12)$$

$$= \text{princarg}[\tilde{\varphi}(\tau R_a, k) - 2\tilde{\varphi}((\tau-1)R_a, k) + \tilde{\varphi}((\tau-2)R_a, k)] \quad (4.13)$$

The resulting product $\Delta f_i(\tau R_a, k)\frac{2\pi R_a}{f_s}$ (that will be denoted as $d\tilde{\varphi}$) is worth careful consideration. From a geometrical point of view, it can be said that it corresponds to the differential angle between the expected (target) phase $\tilde{\varphi}_t$ and the current FFT phase $\tilde{\varphi}$ (tending to zeros during the steady-state and increasing during transients).

Equation 4.13 measures the unpredictability of the signal's features when an attack transient occurs. Settle and Lippe [SL94] proposed that by thresholding this value, transient and steady-state component separation can be

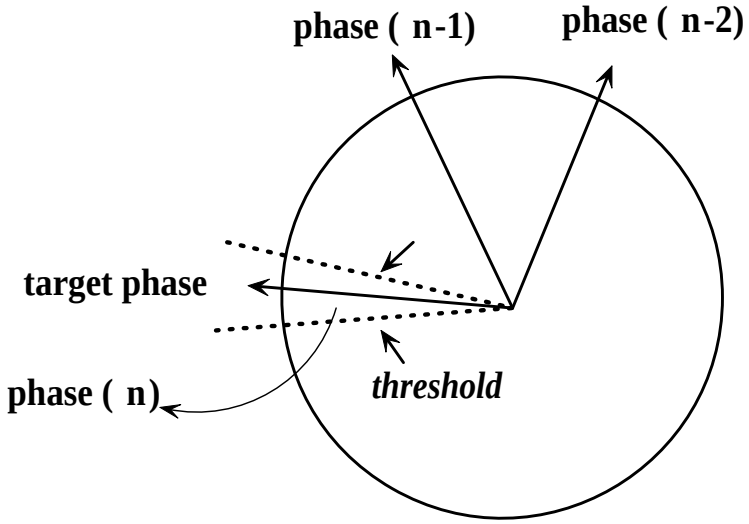

Figure 4.12: Estimation of the phase of the current frame based on the phases of the previous two frames.

achieved. This is shown in Fig. 4.12 where the target phase for $t = n$ is calculated from the previous two frames' phases, and compared with the actual FFT phase. If the current FFT phase is within the thresholded range then it is assumed that the sinusoid is in its steady-state part, otherwise it is considered a transient. Duxbury et al [DDS01] use a threshold that dynamically changes in order to better adapt to the context of the analysis.

In our system, the distribution of the frequency-domain data produced by Eq. 4.13 is analysed for all $k = 0, 1, \ldots, N-1$ (where N is the FFT window length). This analysis is done in a non-cumulative frame-by-frame basis. Standard statistical analysis methods are used for this task.

4.3.2 Statistical analysis

Given a set of observations, the probability distribution function (or simply distribution function) $D(x)$ describes the probability that an observation X is less than or equal to a number x [Wei98]. It is defined in terms of a continuous probability density function (PDF) $f(x)$ by:

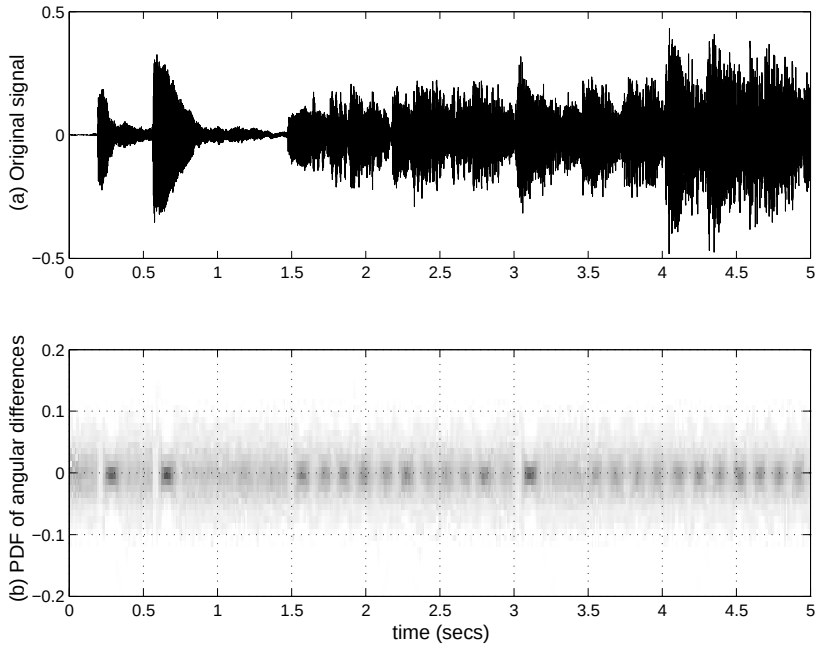

Figure 4.13: Piano signal (a) and its corresponding sequence of probability density functions $f(x)$ of bin-by-bin angular differences (b).

$$D(x) = f(X \leq x) \equiv \int_{-\infty}^{x} f(x')dx' \tag{4.14}$$

or in terms of a discrete PDF as:

$$D(x) = f(X \leq x) = \sum_{X \leq x} f(x) \tag{4.15}$$

Hence, when $f(x)$ exists, it is defined as the derivative of the distribution function $D(x)$.

Let us consider the bin-by-bin angular changes in one frame as our set of observations X. The possible values for x will fluctuate within the $[-\pi, \pi]$ range, as they correspond to our "random" phase difference values. A histogram with grouping interval close to zero can be generated with the data-

set. This allows the study of the probability density function $f(x)$ that describes how our bin-by-bin angular differences are distributed along the phase range. Figure 4.13 depicts an audio signal and its corresponding sequence of frame $f(x)$ along the time axis. A sequence of cross-sections (vertical slices) of Fig. 4.13(b) around its second attack transient ($t = 0.6$) is shown in Fig. 4.14. These figures are used to exemplify a behavioural pattern found in large groups of musical signals.

It can be observed (Fig. 4.14(a) to (b) and (j) to (l)) that in the absence of attack transients, $f(x)$ closely resembles a normal distribution: unimodal, bell-shaped and symmetrical about the mean. However, when attack transients occur, the shape of the distribution does not correspond to that of a normal distribution. The resulting variations can be summarised as follows:

1. Transients: when transients occur, due to the non-stationarity of the signal, the difference between target and actual angular position increases, thus the data-set becomes disperse across the phase range. The spread causes a slight flatness at the top of the distribution, and a decrease of the height of its lobe. This is illustrated in Figure 4.14(c) and (d), particularly noticeable in the former.

2. Beginning of the steady-state: when the steady-state part of a note begins, target and current angular position become closer. The distribution presents a large concentration of zero-phase values, increasing the sharpness and height of $f(x)$'s central lobe. This peakedness is also known as lepto-kurtosis. The PDF sequence shown in figures 4.14(e) to (i), clearly depicts this behaviour.

4.3.3 Spread and attacks

To understand the behaviour of a distribution, the main features of the data-set can be expressed through the calculation of certain summarising quantities. This model of analysis is particularly useful when the data has a strong tendency to cluster around a central value. Those characterising

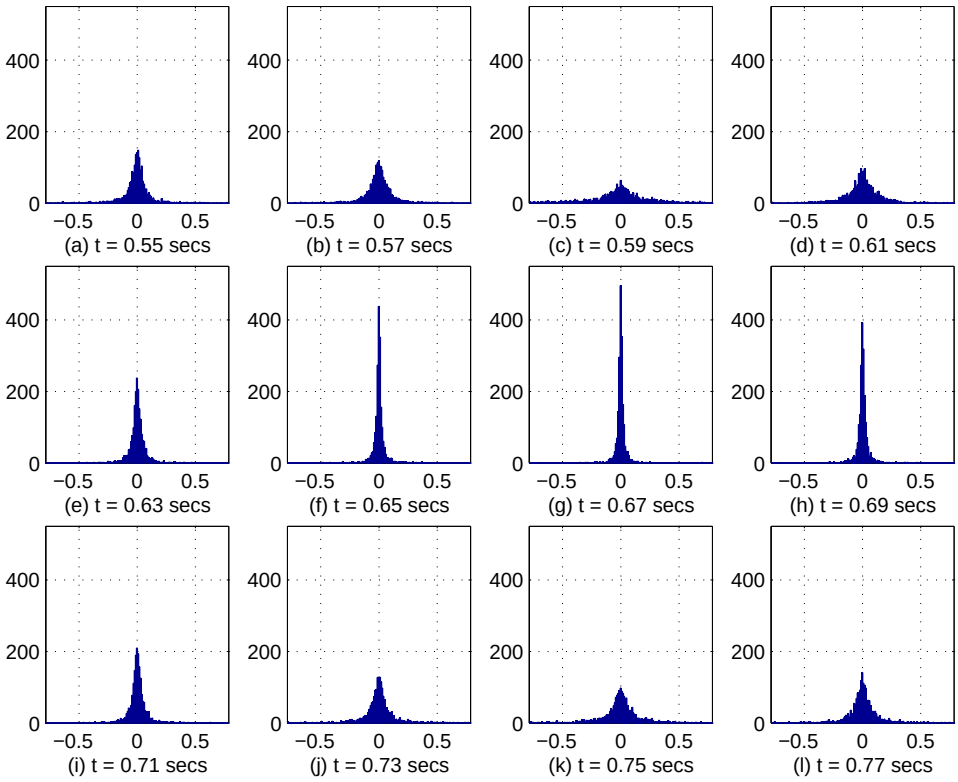

Figure 4.14: Vertical slices of Fig. 4.13(b) around the attack at time $t = 0.6$

quantities are related to the *moments* of the distribution $f(x)$, that can be defined as:

$$\mu_n(a) = \int_{-\infty}^{\infty} (x-a)^n f(x) dx \tag{4.16}$$

where $\mu_n(a)$ is the n^{th} moment about a point a. The raw moments are those calculated about $a = 0$. The first raw moment is known as the *mean* and it is denoted as μ. As our data is centred around zero (or a very close value), the mean does not contribute significantly to our analysis. However, the mean proves immensely useful for the calculation of higher-order moments. These, as we are about to see, provide important information for our analysis. By selecting $a = \mu$ we generate the so-called central moments, defined as:

$$\mu_n(\mu) = \int_{-\infty}^{\infty} (x-\mu)^n f(x) dx \tag{4.17}$$

The second central moment $\mu_2(\mu)$ yields the average of the squared differences between each value in the data set and the mean of the population. It measures the spread of the values against μ. It is known as variance and denoted as σ^2. Due to the squaring process, it is never negative. By calculating the square root of the variance, the standard deviation σ is obtained:

$$\sigma = \sqrt{\mu_2(0) - \mu^2} \tag{4.18}$$

where $\mu_2(0)$ is the second raw moment. The standard deviation is a common measure of the dispersion of the signal. It is widely used when not exposed to extreme values in the data set. The standard deviation of our signal's $f(x)$ is shown in Figure 4.15(b). It characterises onsets as sharp-peak / deep-valley pairs (due to the PDF shape variations). However, as can be seen in the figure, the standard deviation presents a noisy profile that might affect the accuracy of our detection.

An alternative measure of spread can be obtained by calculating the interquartile range (IQR). To calculate the IQR the data-set is divided into two equal-sized groups at the median (approximately equal to μ for normal distributions). The medians of both, low and high, groups are calculated and denoted Q_1 and Q_3 respectively. Then the interquartile range can be obtained as [Wei98]:

$$IQR \equiv Q_3 - Q_1 \tag{4.19}$$

Figure 4.15(c) shows the IQR of the test $f(x)$. The profile is noticeably less spiky than its predecessor. The IQR is apparently less sensitive to small variations in the distribution's spread than the standard deviation, allowing

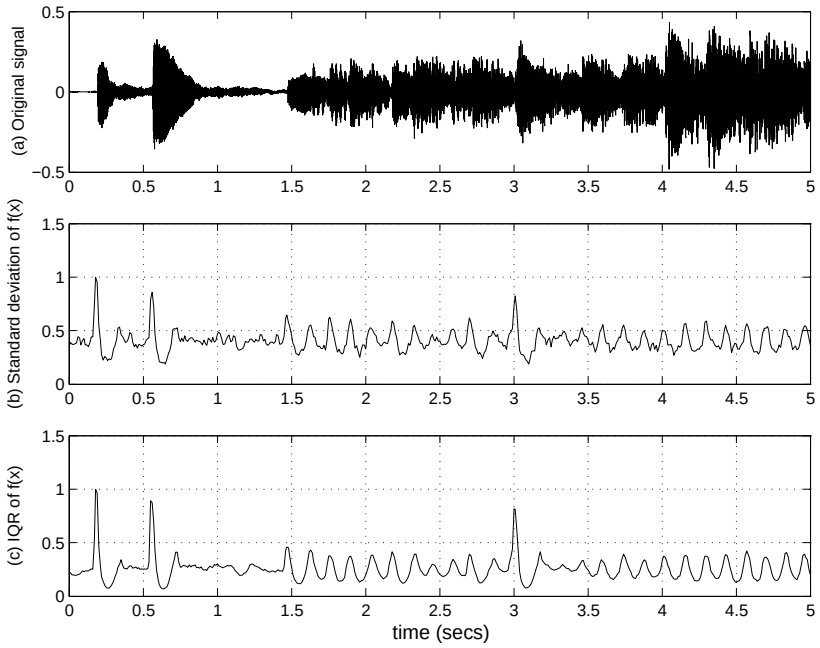

Figure 4.15: Measures of spread of the test signal's PDF: using standard deviation (b) and inter-quartile range (c).

a cleaner recognition of sharp-peak / deep-valley pairs. These observations are consistent between several test audio files. For this reason, the use of IQR is favoured for our implementation.

In real instrument sounds, phase misalignment between composing sinusoids of a given note can cause an spread distribution of $f(x)$ values in the phase range. This is particularly critical when a note evolves for a long time, and the lack of synchronicity between partials becomes evident. In this case, the measure of dispersion is not entirely reliable for onset detection. To boost the robustness of our detection, we intend to complement the analysis of the distribution's spread with the analysis of its shape. Therefore, we will identify the steady-state part of the signal that follows the occurrence of a new event.

4.3.4 Higher-order moments

The third central moment $\mu_3(\mu)$ can be used to measure the skewness of our distribution. As our data is symmetric about the zero-phase value, this is not particularly significant for our analysis. On the other hand, the fourth moment proves very useful for the detection of the beginning of the steady-state part of a note. By normalising it, the kurtosis coefficient γ_2 can be obtained. Kurtosis provides information about the shape of a distribution. It quantifies the flatness or peakedness of the distribution in relation to a normal distribution.

Like the skewness, it is a non-dimensional quantity (i.e. cannot be affected by either additions or multiplications on the data-set). The most common measure of kurtosis, the Fisher kurtosis, is defined as:

$$\gamma_2 \equiv \frac{\mu_4(\mu)}{(\mu_2(\mu))^2} - 3 = \frac{\mu_4(\mu)}{\sigma^4} - 3 \tag{4.20}$$

According to this definition, kurtosis values are zero when the distribution is normal. This only applies for symmetrical and unimodal distributions such as ours. Its behaviour can be roughly summarised as:

$$\gamma_2 \begin{cases} < 0 & \text{If } f(x) \text{ is flat (platy-kurtic). See Fig. 4.16(a)} \\ = 0 & \text{If } f(x) \text{ is bell-shaped (normal). See Fig. 4.16(b)} \\ > 0 & \text{If } f(x) \text{ is sharply peaked (lepto-kurtic). See Fig. 4.16(c)} \end{cases}$$

The kurtosis response is well-adjusted to our needs. At the beginning of the steady-state of a note, the phases of fundamental and partials are locked. The stationarity of the signal allows predictability of the sinusoidal components. The difference between target and actual phases becomes minimal. Thus, the population concentrates close to the centre of the distribution (increasing the sharpness of its lobe).

The kurtosis successfully represents this characteristic as can be seen in Figure 4.17(b). Predictably, deep-valleys in the standard deviation relate to

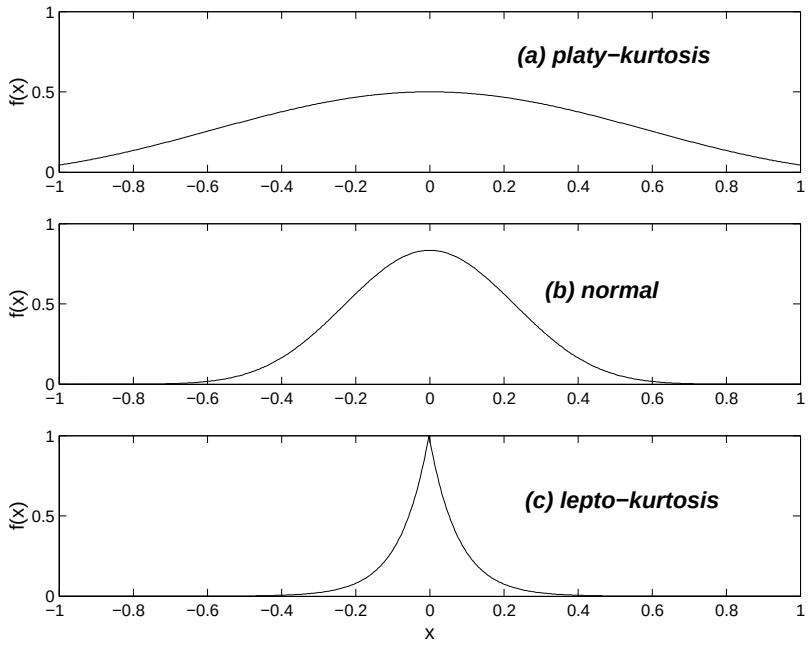

Figure 4.16: Shapes of the PDF for kurtosis analysis

the beginning of the steady-state, as deep-valleys in the kurtosis profile appear during transients. In either case, the valley profiles are too ambiguous to rely on them. We propose that by detecting peaks both in the IQR and kurtosis profiles we will locate pairs of disperse-leptokurtic distributions, hence correctly identifying onsets.

4.4 Peak-picking

As can be seen in Figures 4.15 and 4.17, the kurtosis profile is sharper than that of the IQR. Therefore we favoured an approach where peaks are detected over the kurtosis profile and then matched to the closest preceding peak in the IQR profile.

A peak-picking algorithm is implemented that selects peaks above a dynamic threshold. This flexibility enables the system to adapt to the diversity of profiles produced by different recordings, ensembles and styles of music.

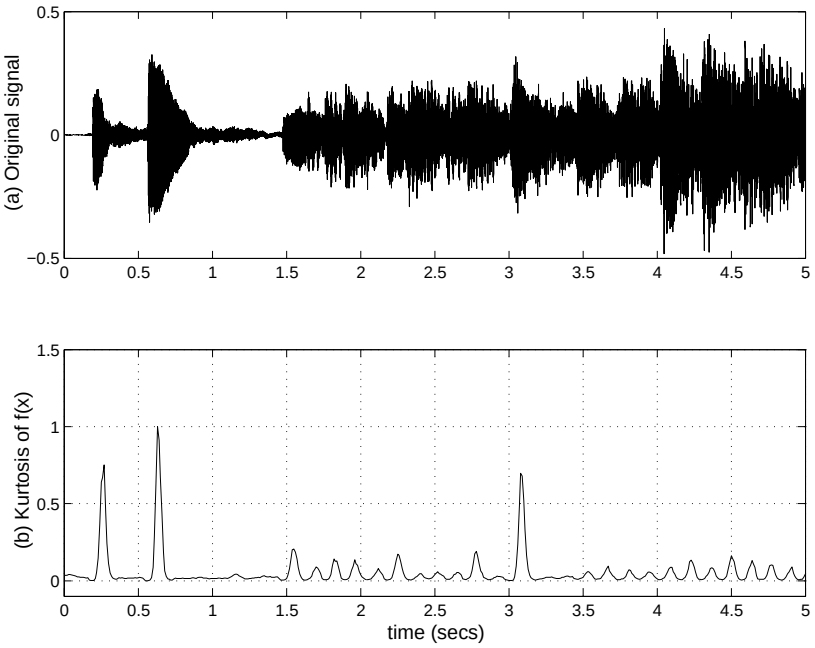

Figure 4.17: Measure of the shape of the signal's PDF using kurtosis.

Thus, obtaining good detections under different contexts is maximised.

Each value of the dynamic threshold δ_t is calculated as the weighted median of an H-length section of the detection functions around the corresponding frame τR, such that:

$$\delta_t(\tau R) = C_t \,\text{median}\,\gamma_2(k_\tau), k_\tau \in [\tau R - \frac{H}{2}, \tau R + \frac{H}{2}] \tag{4.21}$$

C_t is a predefined weighting value. Low values of C_t imply a decrease of the threshold profile. This increases the number of detections, both correct and false. Inversely, high values of C_t make the system more strict by decreasing the number of detections. Therefore, the selection of C_t determines the performance of the algorithm.

In some cases, depending on the instruments being played, the acoustics or the noise of the recording, the detection functions will show peaks that are sub-products of larger, *real* peaks. This does not require measures such

as low-pass filtering as the profile of the functions is noticeably smooth. However, these undesirable detections can be eliminated by supplementing the peak-picking algorithm with an additional routine that evaluates the *minimum distance* between peaks. Correct peaks, need to be separated by this minimum distance, at least, in order to be considered individually. If more than one peak is detected within such an interval, only the highest will be kept as a valid onset.

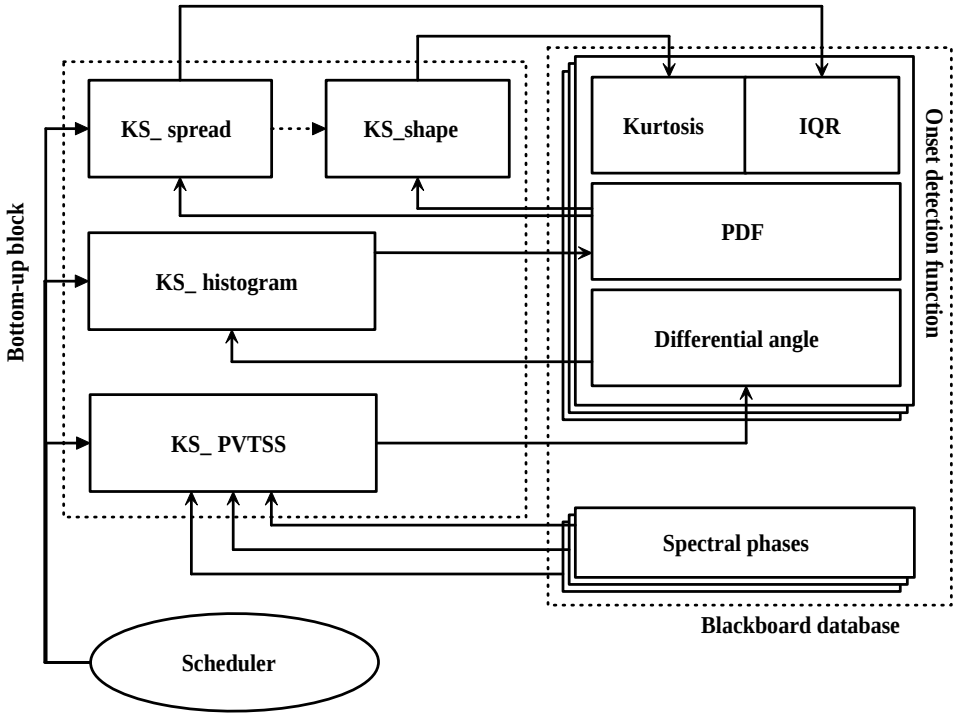

Figure 4.18: Integration of the onset detection into the blackboard framework: a frame's view.

4.5 Integration into the blackboard framework

The onset detection system provides all information about the timing of events within the signal. Hence it is a key component within the architecture of the proposed transcription system.

Regarding the blackboard database, the operation of the onset detector

is related to three main levels of the hierarchy: the spectral phases provided by the time-frequency representation, the onset detection function generated by this algorithm, and the onset hypotheses.

Due to the underlying calculations needed for detecting onsets, the *onset detection function* is a complex level of the hierarchy. This means that it is divided into several groups or sub-levels of information. These sub-levels are related to intermediate data required within the internal process. This is illustrated in Fig. 4.18. The intermediate levels are: the *differential angle* information, the *probability density function* (PDF), and the frame values for *kurtosis* and *inter-quartile range* (IQR).

The data in the blackboard is generated and evaluated through the action of six knowledge sources from different blocks: four from the *bottom-up* processing block, one from the *temporal* and one from the *evaluation and cleaning* processing block.

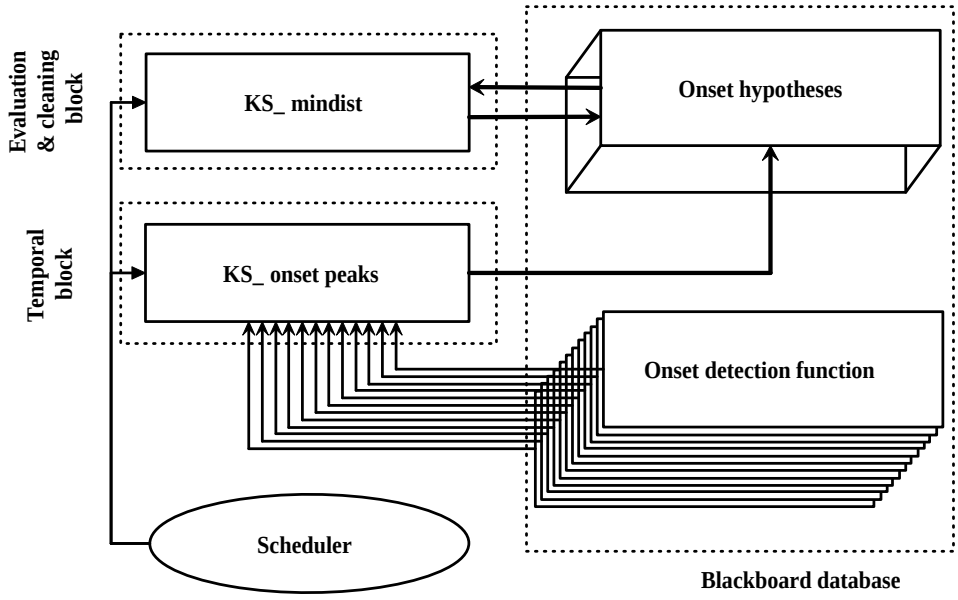

Figure 4.19: Integration of the onset detection into the blackboard framework: a temporal view.

If detecting onsets is high on the priority list, and it is possible to compute them (meaning that there is spectral phase information from the time-

frequency analysis), then the scheduler fires the action of the first knowledge source of the process, **KS_PVTSS**. This bottom-up knowledge source uses the phase of the three last frames to compute the bin-by-bin differential angle according to the phase-vocoder based TSS method explained before. The results are allocated in the first level of the sub-hierarchy related to the onset detection function.

Once this information is available, the scheduler fires **KS_histogram**. This self-explanatory module calculates a histogram based on all frame differential angles. The resulting function is the probability density function of that data, and as such, is placed in the PDF level of the sub-hierarchy.

The third and fourth bottom-up knowledge sources are called, respectively, **KS_spread** and **KS_shape**. Their actions are fired simultaneously by the scheduler to evaluate the spread and shape of the generated PDF and output a single summarising value for the current frame. They modify the *IQR* and *kurtosis* levels of the sub-hierarchy in accordance with the measures they use to perform their analysis.

If enough frame information is collected, the scheduler might decide to calculate the onset time of events within a temporal window. A temporal knowledge source, **KS_onsetpeaks**, uses the peak-picking algorithm and the IQR and kurtosis values of all frames within that window to generate hypothetic note onset times. As mentioned above, it does this by selecting peaks within the kurtosis profile and then finding the times of the closest preceding peaks in the IQR profile.

The final knowledge source, **KS_mindist**, is an evaluation and cleaning module that uses the minimum distance rule to eliminate spurious hypotheses. Figures 4.18 and 4.19 depicts the section of the framework related to onset detection.

4.6 Results and discussion

To test the system's performance a test-bed was created. It consists of a number of short segments extracted from commercial CD recordings. The

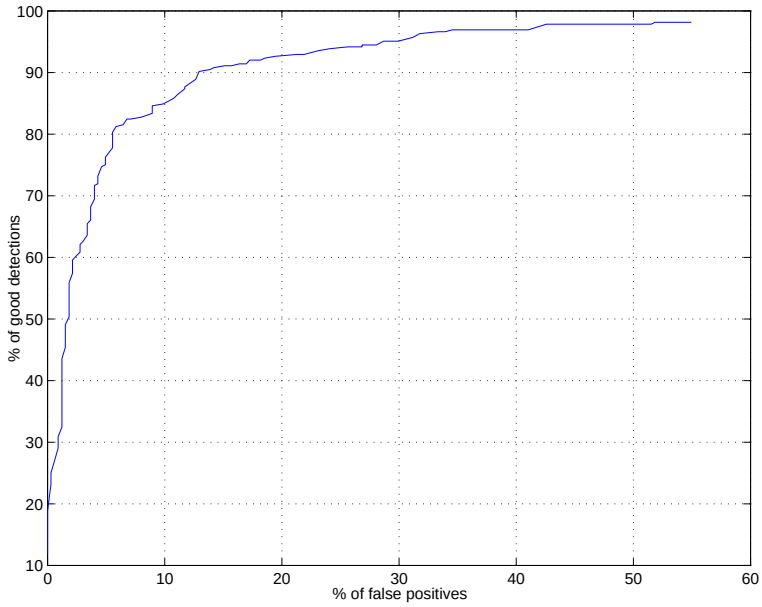

Figure 4.20: Percentage of good detections against percentage of false positives for various C_t values

segments have being hand-labelled, a very time-consuming and unprecise process especially when performed by amateur musicians (such as the author). Onsets were marked by listening to the segment at half the original speed. Then, a train of clicks was produced at the estimated onset times and mixed with the original sound. From hearing this mixture, apparent errors were fixed and a final onset list was produced. This is extremely complicated in polyphonic and complex mixtures.

There are 334 onsets in the test-bed, both in solo performances (183) and in complex mixtures (151). They correspond to different styles of music broadly classified as follows: solo violin, solo piano, solo tabla, jazz trio, pop-rock and pop-rock with vocals. Only the violin and the tabla recordings are monophonic, and in the case of the former reverberation produces a false polyphonic environment. They are all complex tests, usually at high speed, chosen as *interesting cases* for the onset detection to be tested at. Correct

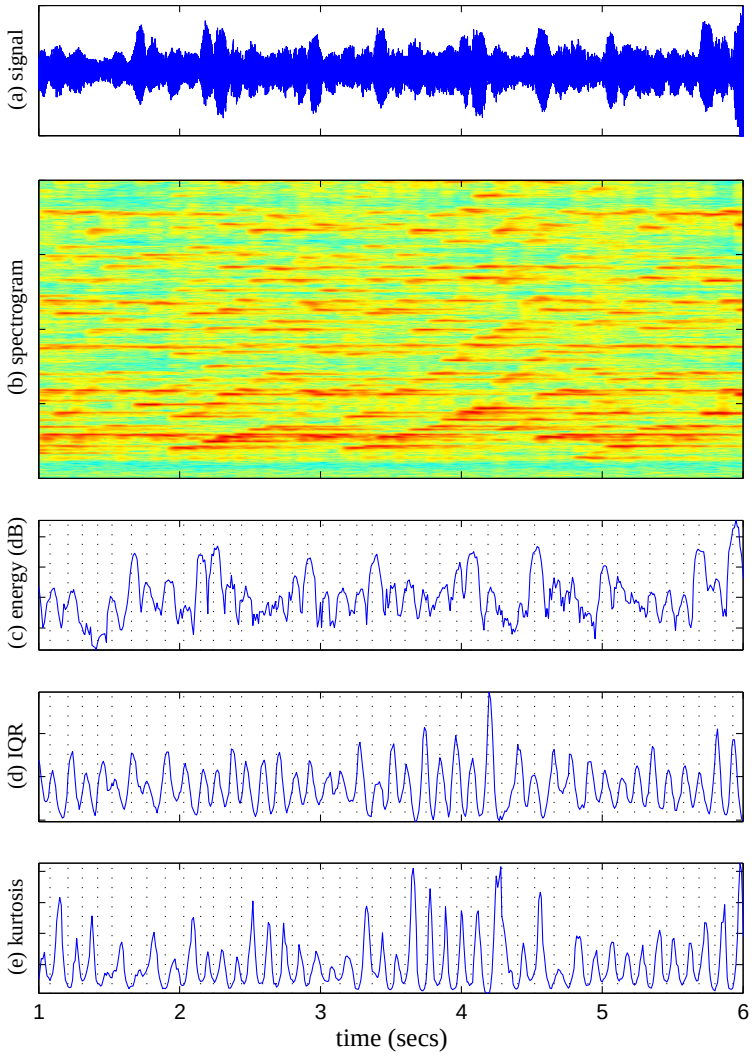

Figure 4.21: Onset detection of a violin signal: the original signal (a), spectrogram (b), energy profile in dB (c), IQR (d) and kurtosis (e). Target onsets are marked by dotted lines.

matches are those when target and detected onsets are within 50ms of each other. This window length copes with the inexact hand-labelling process rather than compensating for the errors of the detection.

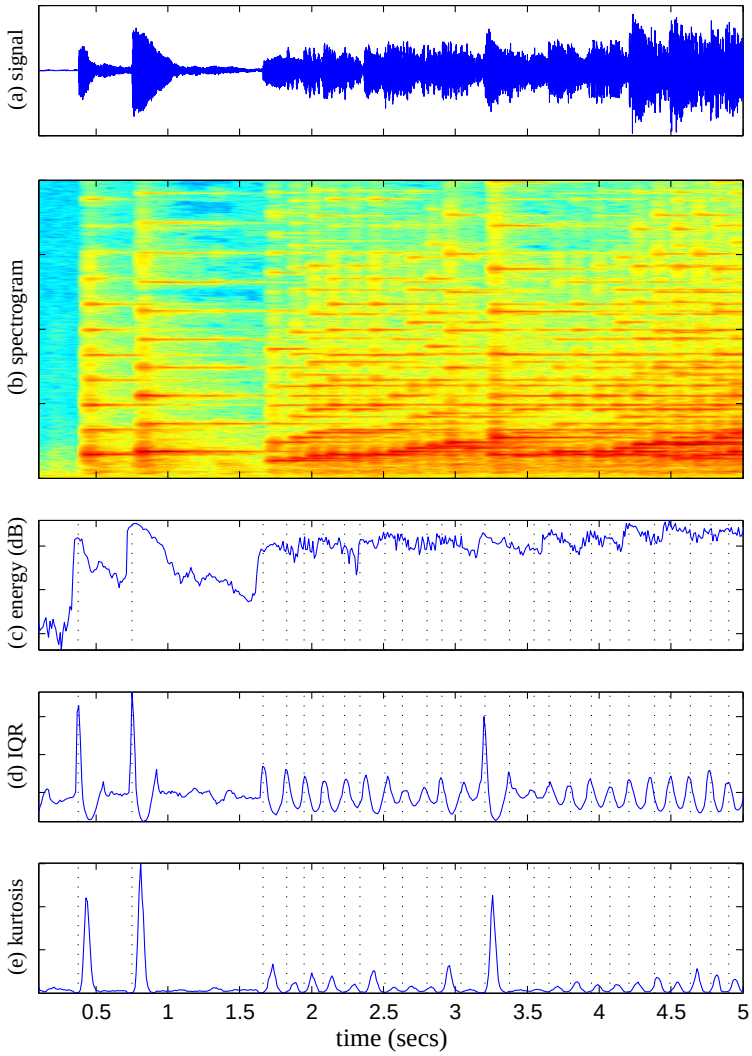

Figure 4.22: Onset detection of a piano signal: the original signal (a), spectrogram (b), energy profile in dB (c), IQR (d) and kurtosis (e). Target onsets are marked by dotted lines.

Initially, we aim to find the optimal value for C_t and to evaluate the capabilities of our detection function. For this purpose all samples within the test-bed are evaluated for varying values of C_t. The idea is to maximise

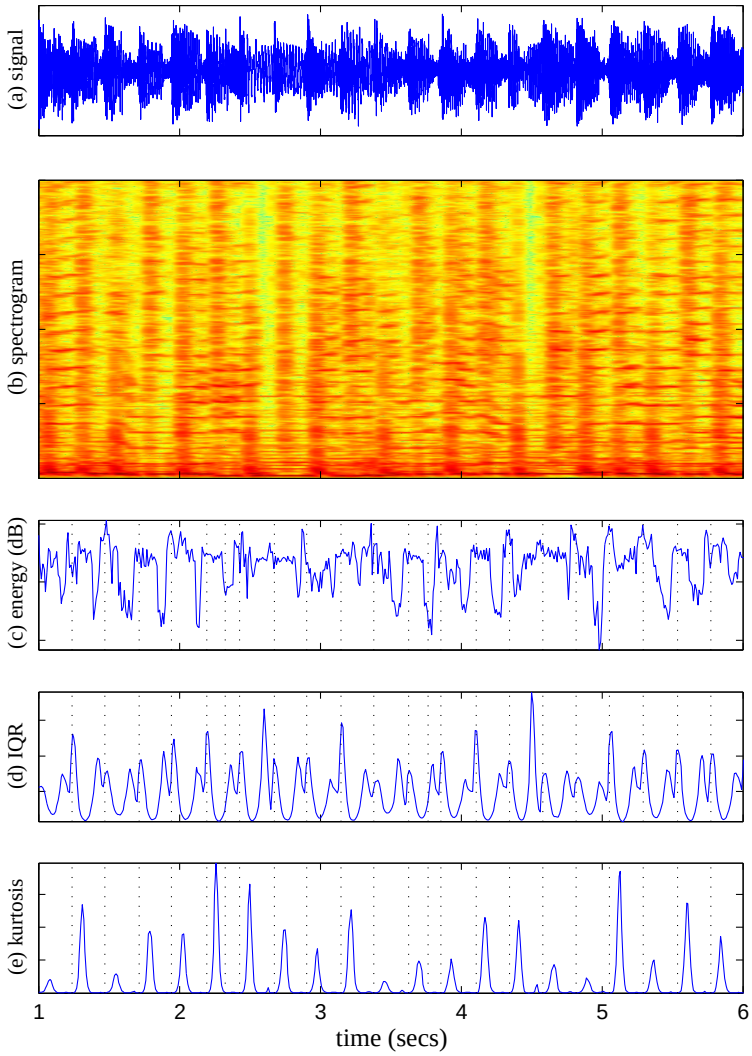

Figure 4.23: Onset detection of a pop signal with vocals: the original signal (a), spectrogram (b), energy profile in dB (c), IQR (d) and kurtosis (e). Target onsets are marked by dotted lines.

the relationship between good detections and false positives (events detected that are not present) for the complete test-bed. This is shown in Fig. 4.20 where the percentage of good detections is plotted in the y-axis, against the

STYLE	% GD	% FP	% FN
solo violin	81.72	3.80	18.28
solo piano	91.83	6.25	8.16
solo tabla	92.68	7.32	7.32
jazz trio	84.85	9.68	15.15
pop rock	87.50	16.95	12.50
pop rock + vocals	90.32	26.32	9.68

Table 4.1: Onset Detection Results

percentage of false positives for different values of C_t.

It can be observed that high rates for good detection (80-90%) are obtained at a cost of around 10% rate of false positives. This is very high when considering the variety and complexity of the signals involved. A system tuned for a specific type of music might generate even better results.

This figure represents the possibilities within the detection functions. It is clear that close to 100% of all onsets are related to peaks in the kurtosis and IQR profile as can be concluded from the high-end of the plotted curve. However, there are peaks (the ones not selected while keeping the trade-off between good and false detections) that can be confused with peaks produced by the interaction between instruments and the presence of large values. These account for the high-percentage of false positives on that side of the curve. Inversely, at the low-end it can be seen that by constraining the peak selection, both errors and good detections are minimised. This, *strict* case, might be useful for certain applications.

The optimal value of C_t is the one corresponding to the "elbow" of the curve. There, the ratio between good and false detections is maximised. This is the value selected for our next set of experiments.

Having selected an optimal value for C_t, tests are performed over the sample base. Table 4.1 shows results grouped according to style. Numbers in the table correspond to good detection (GD), false positives (FP) and

false negatives (FN) rates for each case.

It can be seen that detection rates are high for solo instruments, with a relatively low cost in false detections. The violin produces many *soft* onsets, as it is a non-percussive instruments (i.e. low amplitude, double peaks in the function profile). This implies that the number of false negatives increase as these detections are more difficult to discern. Actually, this is consistent with the low rate of false positives, very difficult in an instrument prone to produce vibrato. An example of the detection functions and detected onsets in a section of our violin sample can be seen in Fig. 4.21. Note the comparison with the energy profile.

On the other hand, better results are obtained with the piano and tabla samples where the maximisation of good results and the minimisation of false positives is achieved. In the case of the piano, this is very encouraging as providing onset information for piano transcription is the main purpose of the current system. Fig. 4.22 depicts energy, IQR and kurtosis of a section of a piano recording (detected onsets are superimposed as dotted lines). Note the noisy energy characteristic as opposed to the the smooth kurtosis profile.

However, as the complexity of the signals increase (i.e. jazz and pop music with vocals), high detection rates are accompanied with higher rates of false positives. Here, the capabilities of the system are tested to its limits. The complex mixture of different sounds, the use of voice and of processing effects (common practice in popular recordings), distorts the phase information of the signal. The system completely depends on accurately obtaining the phase information of the signal, thus phase distortion can badly damage the prospects of good detection. A particular case, that of pop rock music with vocals, is singled out as the worst scenario. The features of the sung voice create a number of situations when onsets cannot be properly accounted for (i.e. sibilance, the percussive sound of a 't' at the end of a word, etc). The system sometimes fails to resolve these situations. This is a difficulty also present during the hand-labelling of those signals. Fig. 4.23 depicts such a case. Note the complexity in the spectrogram. It is worth

noting from this example, that the mixture between highly rhythmic percussive onsets (from the music beat) and soft onsets (i.e. voice syllables, background chord changes) generates detection functions containing peaks of very dissimilar amplitudes. Hence, peak-picking becomes more difficult. A peak-picking algorithm specifically designed for pop signals could overcome this problem, but that is not within the scope of this thesis.

However, the high values obtained for good detection indicate that even with the problems that very complex mixtures imply, the system is successful at selecting onsets. Results show that the uncertainty the framework needs to deal with is minimal for onset detection.

4.7 Summary

Initially we defined note onsets as the beginning of the attack transients of notes. Attack transients are areas of short duration, where energy increases abruptly, where elements are chaotic, thus unconnected from previous and current ones, and which are followed by the steady-state part which is stable and highly predictable.

Several methods have been proposed in the literature based on the analysis of local energy, on energy in the frequency domain, on modelling of the signal into its composing elements and on the statistics of audio signals. Here, we propose an alternative to these methods by relying on the phase of music signals for our analysis.

Phase vocoder theory is used to estimate each current bin phase from the previous two corresponding phases. Then, by estimating the error between current and estimated phase, a bin-by-bin differential angle is generated. This angle tends to zero when the estimation is accurate and to higher values if not. By calculating the histogram of such values, a frame-by-frame probability density function is generated. It is observed that around onset times the distribution is first sparse and then peaky, following the attack-transient steady-state sequence. The spread and the shape of the distribution are measured using IQR and kurtosis coefficients. Peak-picking

is performed over the obtained detection functions to produce a list of onset times. The system is integrated into the blackboard framework by means of six knowledge sources interacting with three different levels of the proposed hierarchy.

The method allows high detection rates as tested with a database of complex real recordings including both percussive and non-percussive (i.e. violin and voice) instruments. Results are susceptible to phase distortion and could be improved by tuning the system for specific styles of music. The current tuning is optimal for the analysis of piano music, which is the main objective of this thesis.

Chapter 5

Pitch identification

The main element of our simplified definition of a musical note is its perceived frequency or pitch. The combination of this information with the timing of a particular event (reviewed in the previous chapter) provides, for most music, an unambiguous description of what is being played. Without aiming to downplay the importance of dynamics and ornaments (especially for differentiating between performances and artists), we propose that the retrieval of the perceived frequencies of played notes, or *pitch tracking*, is the core of any transcription system. The aim of this chapter is the exploration of different ideas set to cope with this task.

First, basic concepts are presented and discussed, then, an exhaustive review is made of previous approaches in the literature. New methods are proposed using conventional analysis in the frequency-domain and a time-domain linear additive approach. Results for both methods are independently presented and discussed. Experiments are made with real recordings of piano music.

5.1 Basics

As happened before when defining attack transients, providing an exact definition of pitch is not an easy task. Most researchers seem to agree that pitch is an auditory phenomenon related to the frequency of a simple tone,

or *tonal height* of a sound, allowing listeners to classify it within a scale from *low* to *high*.

In this context, pitch is a subjective attribute. It depends on the perception of the receiver (see [Bre90, Deu82]). However, we aim to provide an engineering solution to the problem of polyphonic music analysis. As we have no means of measuring the listener's subjectivity we are in need of an alternative view to the concept of pitch.

We will approach pitch as the tonal frequency that provides the better fit between a recorded tone and the note in the *score* that originated it in the first place. Note that we are assuming the existence of a score-audio process before our audio-score analysis stage. This is not necessarily true for all music but is a helpful assumption to make if we are to return a description of music which is consistent with the semantics of music. For example, in this context, an octave interval is clearly composed by two different tones at different pitches, although perceptually the difference might not be so clear. Results will be evaluated against scores, not against an individual's perception of the signal (although this choice is probably fairer).

There are two major cases that can be identified for the pitch recognition problem: the monophonic case, when only one note is present at a time; and the polyphonic case, presenting a multiplicity of pitches and, possibly, timbres. The analysis of the later case is known to be considerably more complex than the analysis of the former. In fact, monophonic pitch estimation is widely considered as a solved problem. Hence, it is in proposing alternatives for the analysis of polyphonies that the work here presented concentrates.

Like its counterpart in the real world, the recognition of notes within a complex mixture hugely depends on the levels of training and knowledge that the "listener" possesses: the more complex the musical input becomes, the more acute the need for prior knowledge. This knowledge can appear in the form of, for instance, rules based on the observation of polyphonic signals, musical knowledge or explicit models of the instruments being played.

Monophonic analysis methods focus on the retrieval of features (amplitude, frequency and phase) from the composing sinusoids of a signal. Unfortunately, such techniques are not applicable to the polyphonic case, encouraging researchers to propose novel ways to interpret the available information. In the following sections, an exhaustive review is made of polyphonic pitch estimation methods in the literature, and their ways of obtaining and using these different forms of knowledge.

5.2 Previous approaches

The first systems set to cope with the polyphonic note identification task were proposed in the second half of the seventies ([Moo75, Moo77, PG77]). However, despite this early start, it was not until the second half of the nineties that a wealth of ideas irrupted into this field, boosted by new discoveries about the physiology and psychology of human perception. Although no system up to this day can claim total success, recent advancements suggest that solutions to subsets of the problem may be closer, providing alternatives for many practical applications. In order to review proposed approaches, they have been classified as belonging to one of four groups: clustering or grouping methods, methods that use external knowledge, optimal representation methods and statistical methods. In the following subsections these methods will be reviewed, stressing their strengths and weaknesses.

5.2.1 Clustering or grouping methods

Here are included all methods that analyse incoming data (from the time-frequency representation) clustering together pieces believed to belong to a single object. This is performed in a bottom-up fashion. The clustering method, and the definitions of these "single objects", varies from system to system. This is the most common approach for polyphonic note identification systems.

Moorer [Moo75, Moo77] developed a system for the transcription of

duets. It presented strong limitations on the frequency range, timbral differences between the two instruments and the intervals that could be transcribed due to overlapping harmonic information. It used a comb filter to identify periodicities in the input signal by minimising the summed absolute value of its magnitude difference. Arguably, it is the first system that produced at least a limited set of results for the polyphonic situation. Around the same time, Piszczalski and Galler [PG77] used an FFT-based approach for detecting strong fundamentals of single instrument polyphonic mixtures. However, their system was basically an off-line monophonic pitch tracking method. Chafe et al, from Stanford University, published a series of papers on the subject [CJ86, CJK$^+$85, CMR82] during the first half of the eighties. Their system used a set of heuristic rules to group peaks at the output of a filter-bank based on the "bounded-Q frequency transform". They experimented with acoustic piano signals but results are not clear enough to assess the success of their ideas.

Maher [Mah89, Mah90] constructed a system for duet transcription. His method for polyphonic note detection was based on the frame-by-frame finding of the pair of fundamental frequencies that minimise the difference between predicted and observed partial frequencies. It used McAulay and Quatieri's sinusoidal model [MQ86] as front-end and a multi-strategy approach for the separation of spectral information that cannot be resolved given the model's frequency resolution. Although limited to certain "ideal" conditions (musically and acoustically), his approach succeeded on real recordings, as long as the voices involved did not cross.

Contemporaneously, Katayose and Inokuchi [KI89], were working in the "Kansei" music system, a system that tried to recreate a human response to music. As part of the system a transcriber was developed for piano, guitar an shamisen (a traditional Japanese instrument). It analysed a time-frequency representation generated using interpolation by complex spectra, a peak extraction method in the frequency domain. Then, it implemented a set of heuristic rules to group these peaks into notes. Tests extended to poly-

phonies higher than two notes but with high error rates. Hawley [Haw93] proposed a system for the transcription of real piano music recordings. His system relied on the use of the STFT, and implemented analysis techniques such as high frequency content for onset detection and spectral comb filtering for polyphonic note identification. However, assessing the success of this system is not easy as tests are not extensive.

Fernández-Cid and Casajús-Quirós [CQFC94, FCCQ98] performed a series of pre-processing and validation procedures into the time-frequency data (obtained by means of a multi-scale sinusoidal model), finally generating a synthesised spectrum where the amplitudes of peaks are relative to a quality-of-fit measure. The synthesised data is then analysed in search of prominent comb partial patterns. Selection and validation is performed following a set of heuristic rules in the frequency and in the time domain. Experimental results are not provided. Another approach is suggested by Dixon [Dix00a, Dix00b] for the transcription of piano music. From the reading of phase-vocoder generated data, time-frequency energy atoms are constructed. Significant atoms are used to build frequency tracks along time. Then, heuristic rules are used to find the set of fundamental frequencies that better explain these tracks. These rules are related to a generic model of musical tones. The system provides encouraging quantitative results when tested with synthesised piano music.

Goto et al [GH99, Got00, Got01] proposed a system for the estimation of bass and melody lines from CD recordings (see Fig. 5.1). Their system is based on several assumptions: the presence of a harmonic structure for bass and melody lines (that do not necessarily include the F0 -fundamental-component); the predominance of the bass line harmonic structure for the low frequencies and of the melodic line harmonic structure for middle and high frequencies; and finally, the continuous temporal trajectories of those lines.

The system starts by calculating the instantaneous frequencies for the STFT filter outputs, and then extracting candidate frequency components

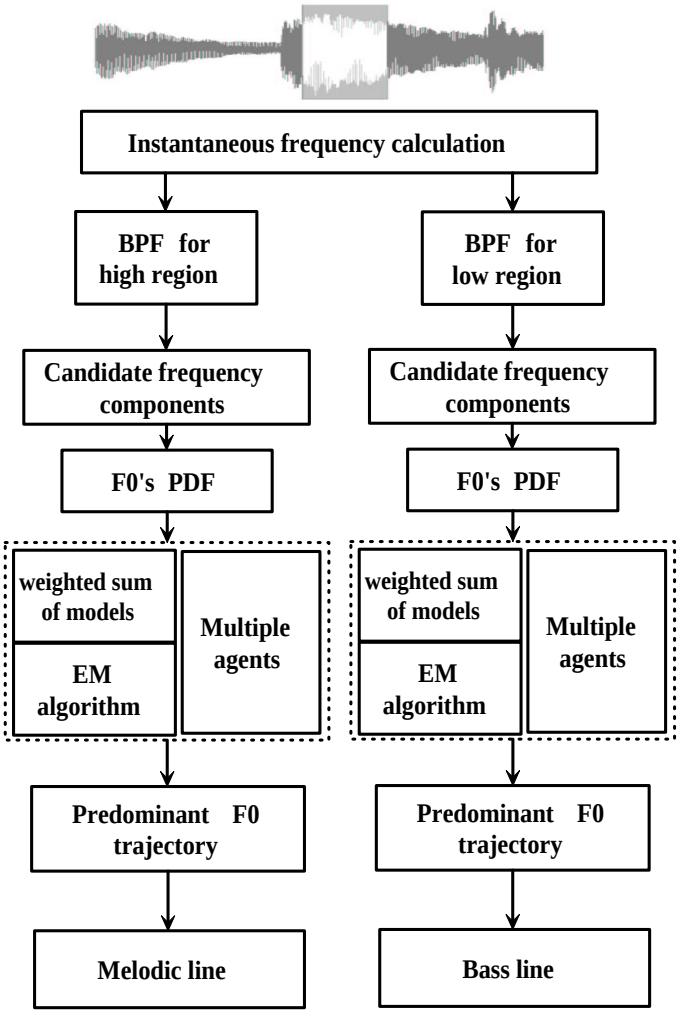

Figure 5.1: Block diagram of Goto's algorithm for the estimation of melodic and bass lines.

from those instantaneous frequencies. Then it splits the spectrum of the signal into two regions: a low and a middle/high region. This is done by means of bandpass filtering. It defines a so-called probability density function (PDF) of the F0 for each region, constructed from the filtered frequency components. The main idea is to assume that the observed PDF is the weighted sum of tone models (a tone model corresponds to the PDF of a typical harmonic structure for a given F0 within that region). The sum's

weights are used to construct the modelled F0's PDF, as they are relative to the predominance (or dominance) of a particular F0. The estimation of those weights is done by means of the expectation-maximisation (EM) algorithm [Dem77]: an iterative algorithm that computes maximum likelihood estimates from incomplete observed data (such as the observed PDF). Determining the predominant F0 is finding the frequency that maximises the modelled PDF. Finally, a multiple-agent architecture is used to track different temporal trajectories of the F0. This compensates for instability in the frame-by-frame analysis output.

The system has proved successful within its defined scope, the assumptions made being effective for most commercial CD recordings. By limiting their scope, the authors avoided dealing with some of the most complicated problems of the multi-pitch estimation task. Nevertheless, it stands as one of the very few systems that provides a viable and robust solution for a particular problem within very complicated sound mixtures.

Klapuri and his group published a corpus of research [Kla98a, KVH00, Kla01a, KESV01] intended to develop a full polyphonic transcription system. Their research topics include onset detection, beat tracking, multi-pitch estimation, source separation and the integration of the developed algorithms. We mention only the proposals relevant to polyphonic note identification. Klapuri's multi-pitch estimation method is organised as follows: first, predominant pitch estimation is performed in the pre-processed spectrum, then, the detected sound is linearly subtracted from the mixture. The process is repeated iteratively until an estimated number of voices are retrieved. The corresponding block diagram can be seen in Fig. 5.2.

The pre-processing stage consists of a noise-suppressing algorithm that removes all non-harmonic and non-melodic components (including the sound of drums and percussive instruments) independently on each frame. Details can be found in [KVES01]. To calculate the predominant pitch of the processed spectrum, the frequency-domain signal is divided into multiple logarithmically-separated frequency bands (simulating the human hearing

process). At each band b, $b = 1 \ldots B$, a likelihood vector L_b is calculated. The processed spectral samples $S_p(k)$, are in the range $k = k_b \ldots k_b + K_b - 1$ (K_b is the number of samples in the band). The bandwise F0 likelihoods $L_b(k)$ are calculated by finding a series of spectrum samples that maximise them:

$$L_b(k) = max_{m=0\ldots k-1}[N_H \sum_{h=0}^{H-1} S_p(k_b + m + hk)] \tag{5.1}$$

where $m = 0 \ldots k-1$ is the offset of the series of partials corresponding to F0, $H = (K_b - m/k)$ is the number of partials in the sum and N_H is a normalisation factor.

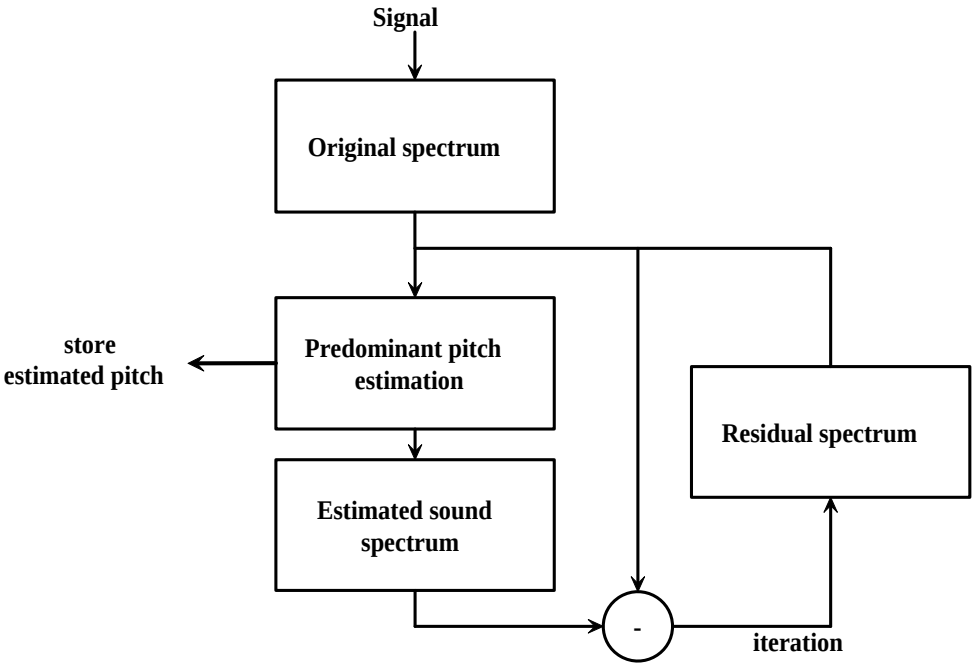

Figure 5.2: Block diagram of Klapuri's iterative multi-pitch estimator.

Finally, bandwise likelihoods are combined (following heuristic rules) to generate a global estimate. The algorithm returns the fundamental frequency of the estimated note, its associated inharmonicity factor and the complex values of the harmonic series of the sound. With this data the

spectrum of the detected note is estimated, using a "spectral smoothing" algorithm [Kla01b] (that attenuates the effect of removing shared -harmonic- information between notes in the mixture), and then linearly subtracted from the mix. The complete process is repeated on the residual signal until a series of clues retrieved during the process [KVES01] indicate that all voices have been detected.

The system shows considerable success in frame analysis even with high polyphonies, multiple timbres and complicated acoustic conditions. However the lack of temporal integration in most of the experiments cast doubts about the efficiency of the system when faced with real musical recordings. In recent publications, temporal processing modules have been added to deal with this, however no quantitative results are provided, and qualitative results seem to indicate that detection rates become considerably lower than the frame-by-frame case. In any case, the wealth of ideas presented during the length of this research has contributed significantly to the advancement of the field.

An alternative for the grouping of information is proposed by Von Schroeter [vS00, vSDR00]. His system uses an auto-regressive model to represent the musical signal and a connectivity approach to relate points across time (as part of spectral lines across neighbouring frames) and points across frequency (as part of the frame's patterns). Spectral lines are related to each other and used to support F0 hypotheses. The approach has the advantage that no synchronicity is required between the note's components, eliminating the need for robust frame-by-frame detection. However it is prone to errors due to inharmonicity and to discontinuities on the spectral lines of certain notes (causing multiple detections). No extensive testing is provided. This is similar to the method proposed by Martins [MF02] who segregates the trajectories of harmonic structures in time, eliminating spurious trajectories caused by harmonic relationships. Trajectories are selected or rejected according to a probabilistic criteria. The system shows mild success when tested using synthesised simple sounds under ideal conditions.

5.2.2 External knowledge

A different kind of system tries to use external high-level information related to what we might expect to find in the musical structure of the signal. In this approach the main idea is to find similarities between patterns in the analysed signal and patterns in the external source of knowledge. However, high-level information can also include knowledge about the organisation of the musical data or about the semantics that generated this data in the first place.

Inside this category falls Martin's system [Mar96a] for automatic transcription. Its architecture is explained in chapter 2. The hierarchy organises the data in five levels: tracks, partials, notes, intervals and chords. Initially the system used the STFT as its time-frequency representation but an improved version [Mar96b] incorporated the use of the log-lag correlogram (explained in chapter 3) as its front end, as an attempt to provide means for bottom-up detection of highly overlapping intervals by using a more perceptually significant processing scheme. The different "knowledge sources" are built around a set of mostly heuristic rules intended to find connectivity patterns between the different levels of the hierarchy.

One of the most salient points of the implementation is its use of musical knowledge, i.e. that of the construction and detection of intervals. It finds intervals and chords of a certain structure, corresponding to the expected musical characteristics of the analysed music. However limiting the use of these rules may be (i.e. the type of music that can be analysed), they constitute a breakthrough due to their relationship to the way human listeners perform the transcription task. Success, on synthesised piano music, is limited, especially for octave intervals (even with the front end's upgrade).

Kashino et al [KNKT95, KH96, KM98] also made use of the blackboard paradigm for music scene analysis. The architecture, also explained in chapter 2, basically consists of three main blocks: pre-processes, main processes and knowledge sources. In the pre-processing block, the spectrogram is calculated and peaks are selected by means of the pinching

plane method [KNKT98]. Onset times are obtained by using the methods described in [Ros92, DH89]. Frequency peaks are clustered together according to calculated onset times. These clusters are known as processing scopes. In the main-processes block, processing scopes are introduced into a three-hierarchies structure: frequency components, musical notes and chords. Similar to Martin's, knowledge sources in this system are intended to find and evaluate relationships between the different levels of the structure. They are divided into bottom-up, top-down and temporal processing modules. Note that the inclusion of temporal processing modules adds an extra dimension to the connectivity patterns to be found. These knowledge sources include explicit or derived information from databases of recorded songs, chords and instrument tones. Kashino et al performed extensive testing of these databases in order to generate statistical data about the nature of the progressions, chords, intervals and timbres of the music being analysed. They also included music-theory rules to recognise certain polyphonies. All this process required a considerable amount of pre-work and large data-storing capabilities. The system intends to find the optimal set of connectivity patterns that can explain the played music. Results, on resynthesised MIDI files using samplers, show high detection rates. However, it is unclear if sounds used for testing and for the database are the same. No results with real recording conditions are provided.

Rossi et al [RGL96, RGL97] used a database of piano tones to compare the location of the harmonic series with that of the peaks in the analysed spectra. The system makes use of the STFT for the analysis of piano signals. The procedure consists of two parts: an off-line process for the construction of the database and an on-line frame-by-frame identification of notes. For the building of the database, a partial identification procedure is implemented in recordings of individual notes executed in a real piano. Peaks of the STFT are selected or rejected according to comparison against a low-pass filtered version of the spectrum and a set of thresholds. If the identified partial location is consistent along a certain number of time frames, then

it is added to the database (allowing a 1-bin difference window). During the note identification procedure, the incoming signal is analysed using the same procedure, and the retrieved partials' locations are compared against the database (see Fig. 5.3). A note is recognised when a certain amount of its partials are found in the analysed frame. Note that only the location of the partials is considered, amplitude is only used for timing purposes. The authors claim that overlapping problems are resolved due to the inharmonicity of the piano, that creates considerable separation between the location of high-frequency partials of different notes. Experimental results show considerably high detection rates in polyphonies of up to 4 voices, although in somehow ideal conditions. Again, it is not clear if the test piano sounds are included in the database. A similar idea is proposed by Ortiz-Berenguer and Casajús-Quirós [OBCQ02]. They use physical modelling to create a database of the individual notes of a piano. This database is evaluated against a set of recordings of acoustic piano notes. The modelled database is used for the frame-by-frame identification of notes and chords (not continuous music). Their system calculates the inner product between the analysed spectrum and their synthesised book of spectral patterns. The identification of chords is performed by iterative detection of the predominant note and its substraction from the analysis spectra.

5.2.3 Optimal representation methods

An alternative subset of multi-pitch estimation approaches defines the signal's time-frequency representation as the result of a modelling process. In this context, the task of note recognition is reduced to find an optimal set of parameters such that the difference between original and modelled representations is minimal.

Along these lines, Tanguiane proposed a model that makes extensive use of the theories of perception and music [Tan93]. His theoretical findings explained high-level structures (such as intervals or chords) as generated by repetitions and transformations of basic elements (such as tonal structure).

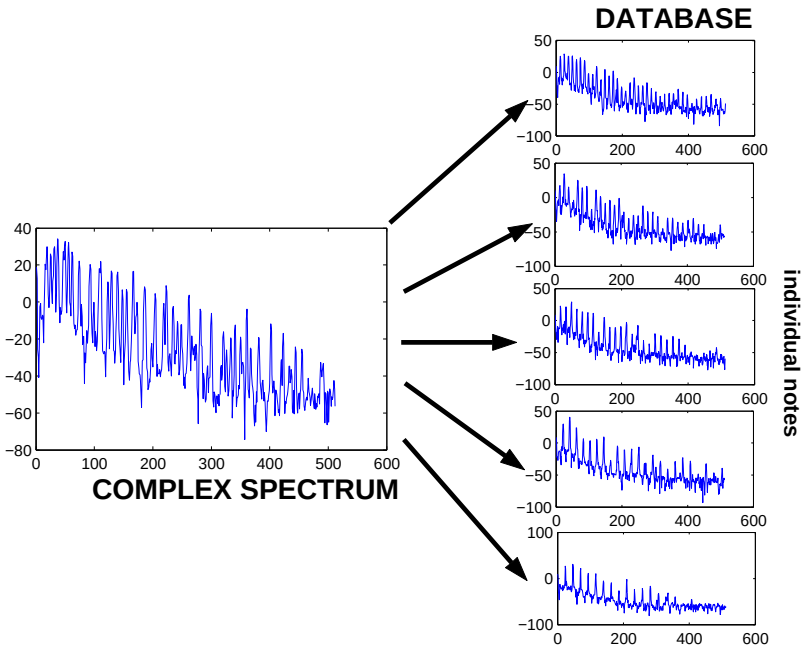

Figure 5.3: Comparison of partial's positions within the spectrum against an spectral database of individual notes, as suggested by Rossi et al.

For instance, Tanguiane explains a chord spectrum as a tone spectrum which is transformed and repeated several times. These transformations are translations along the log-frequency axis corresponding to the musical intervals between the chord notes. The problem is then reduced to find the correct set of transformations that can explain the observation of a particular chord spectrum. This is done by means of a triple recognition process: by finding harmonic intervals within the given chord, by finding melodic intervals between the chord and its predecessor and by finding melodic intervals between the chord and its successor. For this, a "boolean" spectrum of the chord is constructed disregarding the spectral shape of the sound. Harmonic intervals of a chord are recognised by peaks of the autocorrelation function of the chord boolean spectrum. Melodic intervals are found by peaks of the correlation function between the chords boolean spectra. Recognition is granted by satisfying any one of the interval-finding procedures. The

approach was tested using four-voice synthesised Bach music, resulting in considerably high detection rates when high spectral resolution is allowed. However, the acoustic conditions of the test data are arguably too artificial. How the proposed theory deals with real recordings is unclear. It seems to be susceptible to low-frequency noise, reverberation and sounds with missing fundamentals, as it initially relies on the shifting of the analysed spectra by the frequency of the lowest available peak. Also it is unknown how the assumed tonal structure deals with unconsidered properties of real musical sounds, such as inharmonicity. Experimental facts apart, Tanguiane's research breaks usual paradigms of computer music analysis by being able to propose an approach where high-level musical structures can be recognised without complete understanding of their underlying components.

Lepain [Lep99] proposes a system that explains a spectrum as a linear combination of exponential spectrum models. It uses a constant-Q transform as front end and recognises pitch information as a result of a three stage process. First, it determines the "harmonic grid" of the predominant pitch in the mixture. This is done by locating the maximum peak in the spectrum and assuming that the desired harmonic grid belongs to a note with frequency $f_0 = f_{max}/m$, with $m = 1, \ldots, M_{max}$, an arbitrary number. The corresponding comb patterns are generated and then evaluated using a set of heuristic criteria. The second stage tries to estimate the best spectral model for the selected harmonic grid. A model is defined as:

$$M_{\alpha,m}(r) \begin{cases} A\frac{\alpha^r - 1}{\alpha_1^m}, & 1 < r < m \\ A\alpha^{r-m}, & r > m \end{cases}$$

where A is the model's amplitude, r is the rank of a harmonic of the model, m is its order (the rank of the model's maximum) and α is its slope, which defines the shape of the grid. The parameters are estimated from the spectral shape of the harmonic grid selected on stage one. Once the best model is estimated, stage three consists of the subtraction of the model from the original spectrum. The process is then iteratively repeated. Results on

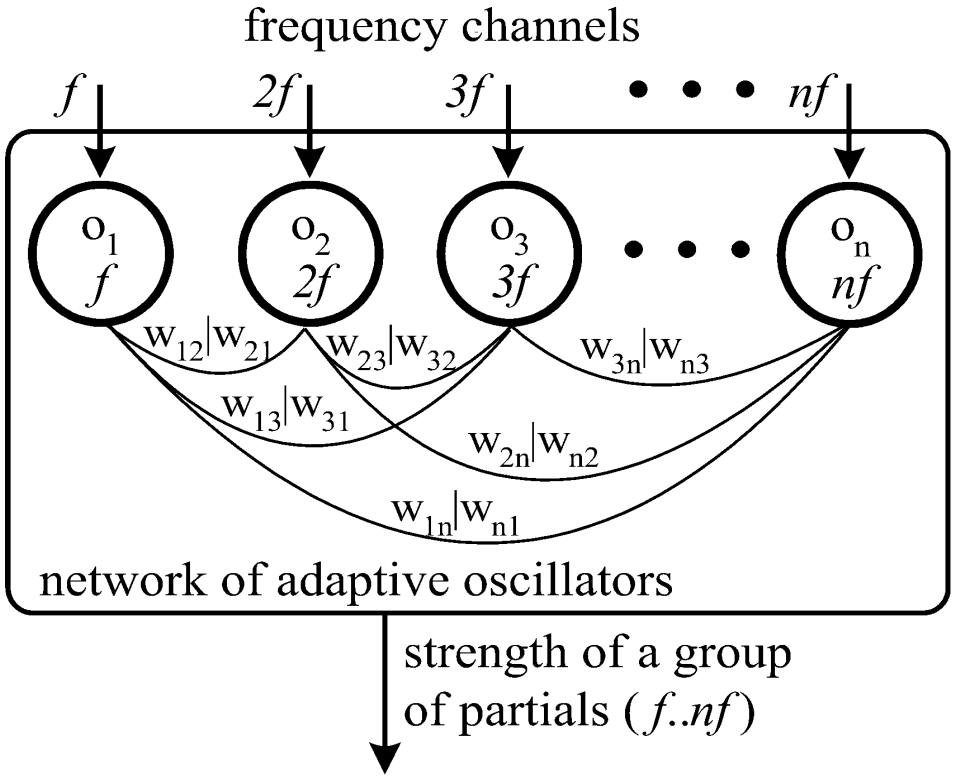

Figure 5.4: Marolt's network of adaptive oscillators. From [Mar01] reprinted with permission.

a limited test-bed showed that the system is not robust when faced with harmonically related intervals (as the harmonic grids share too much information). Also, and due to the greedy subtraction procedure, the detection of low notes during the first iterations eliminates relevant information for the detection of subsequent notes.

5.2.4 Statistical methods

Our final group of methods is constituted by those approaches that use statistical frameworks to explain the behaviour of polyphonic music signals. The idea being that the understanding of the statistics governing music allows the recognition of its semantic structure.

Shuttleworth and Wilson proposed the use of neural networks to pro-

cess the information provided by a multi-resolution Fourier transform-based front end [SW95a, SW95b, Shu96]. The networks were intended to recognise notes from the multi-scale representation. Initially they attempted this using a multi-layer perceptron, trained using the generalised delta rule [RHW86] for the recognition of triads (randomly generated from a collection of real instrument samples). However, after extensive training, the network failed to generalise to unknown triads. A re-branding of the problem presented a spectrum as the sum of the spectra of individual notes, and the transcription problem as an energy minimisation task. Hopfield networks [Hop82] were implemented to find minimum energy states. Different networks are used on different scales of the representation until they all reach stable states. Their output indicates which prototypical spectra (from an arbitrary database) can be used to reconstruct the original spectrum. The selection of the right network on the right scale is based on an error minimisation function. Results show large error rates, mostly due to spurious detections. However, the major drawback of the approach lies on the excessive amounts of information generated by the MFT.

Marolt [Mar98, Mar99, Mar00a, Mar00b, Mar01] assessed the performance of several standard neural network configurations for the transcription of piano music. The signal is processed using an auditory model based on a gammatone filter-bank followed by a model of the dynamics of the inner hair cells [MH91]. The outputs of the model were then processed by networks of adaptive oscillators such as the one depicted in Fig. 5.4. Adaptive oscillators adapt their phase and frequency in response to external input [LK94]. The output of each oscillator indicates the success of the synchronisation with the driving signal. Each oscillator is limited to sync with frequencies within a specified range around its central frequency. Marolt coupled harmonically related oscillators, constructing networks of oscillatory partial detection related to each of the notes of the piano range. The smoothed oscillator-networks' outputs were used as inputs for a bank of feed-forward neural nets trained for the transcription task. Each network

is also related to a note within the range, and they also receive as inputs the amplitude envelopes of the gammatone filters centred at the notes' frequencies. Networks were trained using backpropagation on a large database of piano chords. Results, on real recordings and on synthesised MIDI files, show high detection rates for complex polyphonies. Octave intervals and short notes appear to be the most common source for errors.

Carreras et al [CLL99] propose a well-documented system for chord decomposition. It is conceptually similar to Tanguiane's approach, agreeing with the Gestalt-based idea that extracting global information is generally easier than extracting detailed information. In general they explain chords as composed by a set of basis signals or sub-chords. The decomposition task is viewed as the projection of a given chord into the space defined by the sub-chords. The system first processes the signal through an auditory model [vIM92] representing the nerve patterns of twenty frequency channels on a Bark scale. A "virtual pitch extractor" performs periodicity analysis on these patterns by means of a short-time autocorrelation function. The summary autocorrelation over the 20 channels is termed a *completion image*. So called *chord images* are created by calculating the normalised weighted mean of the completion images in the time interval between consecutive onsets (an onset detection block is used for this calculation). Chord images are mapped onto a two-dimensional self-organising map (SOM) [Koh95] trained to recognise sub-chord images. An intense off-line procedure is performed for the creation of a database of sub-chords, training of the network and labelling of the network activation patterns. From the sub-chords recognised by the SOM and their "ranking" the individual notes of the chord are deduced. Much to their credit, the system is tested with CD recordings of piano and harpsichord performances, containing polyphonies of up to 5 notes (without considering the effect of reverb). Results show that the system is successful in performing harmonic analysis (most fundamental frequencies of the present chords are correctly detected), but presenting errors on the detection of the less predominant notes of the analysed chords. Still, con-

sidering the nature of the test data, it is obvious that the SOM is able to generalise from its prototypical basis to real musical conditions.

Abdallah and Plumbley [Abd02, PAB+01] proposed that notes can be seen as independent components of a spectrum, hence by using independent component analysis (ICA), polyphonic note recognition can be achieved. Moreover, because there is a large number of possible notes but only few of them are used in any given time, sparse coding [FO96], an ICA related technique, can be used to represent spectral information. Note probability is modelled by a distribution containing a sharp peak around zero (depicting the fact that most notes are off during a particular chord). The sparse coder learns about notes and how to detect them in an unsupervised fashion. No prior knowledge is required. The system only considers a one-dimensional representation of the data (frequency). Future improvements may include a system that analyses the two-dimensional time-frequency characteristic of the signal providing more reliable detection. This is expected to increase the complexity of the model. The model provides a robust explanation to the underlying processes in musical signals. Unfortunately, only a few qualitative results are provided on synthesised polyphonic pieces.

Walmsley [WGR99b, WGR99c, WGR99a] developed a parametric model for polyphonic musical signals. It is based on the bayesian methodology, a graphical model used for the encoding and extraction of uncertain knowledge in expert systems, represented as probabilistic relationships between a group of variables [Hec95]. In this approach, data observations are explained by means of a general linear model (GLM), as the weighted sum of a number of notes (basis). Notes are composed by a set of harmonics (with a certain fundamental frequency and an amplitude vector) and are modelled using in-phase and quadrature sinusoidal components. Several frames are considered at the same time to make the detection robust against transients and spurious information. All the involved parameters are estimated jointly using Markov chain Monte-Carlo (MCMC) techniques (which makes for a very slow process). The approach provides a rigourous

framework for polyphonic music signals, however, it is not clear from the results (on synthesised piano samples) how well it performs. Being critical, the assumed model for notes is too simple, considering an amplitude invariant harmonic comb in the frequency-domain, and disregarding transient information. These choices may however be justified by the mathematical complexities posed by the model. Sterian et al [SSW99] also made use of the Bayesian formulation, in this case, for the integration of pre-processed data. The input comes from the use of a modal distribution (explained in chapter 2) to represent the original signal, and the processing of the distribution's data by a Kalman filter, used to track partials' trajectories. The goal of the Bayesian model is to group identified sets of tracks into note events. The model does not have any special consideration for harmonically related intervals, which makes it susceptible to this kind of errors. Detection rates are high for synthesised MIDI files of up to 4-voice polyphony, however, more thorough testing is needed. Hainsworth [Hai01] discusses possible variations on Sterian's method, and the possibility of applying the bayesian methodology directly over a time-frequency representation (such as his proposed reassigned spectrogram [HW01]). However, these are only possibilities for an on-going research project.

5.3 About polyphonic pitch estimation

The problem of polyphonic pitch estimation remains elusive, even after the proposal of so many interesting and well-founded methods. In an attempt to explain this, we will analyse how certain characteristics of polyphonic mixtures affect the detection task.

Almost invariantly, pitch estimation systems rely on the analysis of information in the frequency domain. This is justified as in time-frequency representations, periodicities in time, such as pitch, are represented as energy maxima in the frequency-domain. This suggests that the grouping of certain series of these energy maxima generates patterns or structures that may be related to notes in a music file. Notably, the presence of a note is

specifically associated with the presence of a comb-pattern in the frequency-domain with lobes approximately at the position of the multiples of the fundamental frequency of the analysed tone. As seen before, the recognition of these patterns has been the main objective of most estimation methods. However, relying on the analysis of the frequency-domain data has some disadvantages. For example, the resolution limitations of most time-frequency representations affects the estimation of the "exact" frequencies of spectral peaks. This, already a problem for monophonic sounds, is critical for polyphonies. A number of methods can be used to deal with this (see Brown [Bro93]), an example being the phase-vocoder, as explained in chapter 3.

However, this is a minor problem when compared with certain features of polyphonic mixtures, present even for single instrument recordings. We will expand on the explanation of the two we consider the most relevant:

1. Harmonicity: Two sounds are considered to be harmonic when they have a fundamental frequency ratio $a : b$ (where a and b are positive integers). This implies that every b-th partial of Sound 1 overlaps every a-th partial of Sound 2 [Kla98b]. Intervals of the western musical scale commonly produce perfect or near-perfect harmonic relations (e.g. the fifth - $2 : 3$, the third - $4 : 5$. Note that in equal-tempered instruments, such as the piano, these ratios are not exact. However given the frequency resolution of our analysis these differences are almost negligible). Hence simultaneous notes often generate important overlapping between partials. As mixing is an additive process, sinusoidal features of overlapped partials cannot be obtained from the polyphonic mixture of harmonic sounds. This is illustrated in Fig. 5.5). Harmonicity thereby hugely complicates the process of identifying notes. A common interval, the octave ($1 : 2$), is particularly difficult having every other partial of Sound 1 overlap all partials of Sound 2. The detection of such intervals is a common source of mistakes in reviewed systems.

2. Polyphony: When more than one note is present, peaks related to

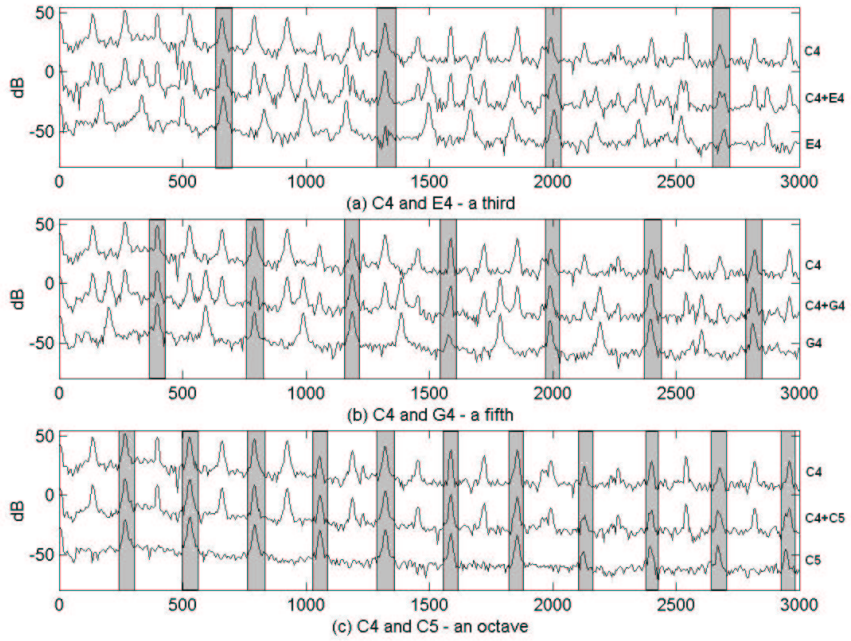

Figure 5.5: The problem of harmonicity with chords made of two notes related by simple intervals. Dark areas indicate frequency regions where partials overlap between the two notes. Top: third interval - Middle: fifth interval - Bottom: octave.

> different pitches can lie in the same frequency bin and therefore are not uniquely identifiable. This problem gets progressively worse as more notes are added. Increasing polyphony often means increasing error rates. The probability of harmonic series extending to common frequency bands will increase proportionally to the polyphony of the song analysed.

However, all mentioned characteristics refer to the spatial-analysis of the signal (in frequency-domain representations of signal's segments), regardless of the time-varying nature of a musical signal (and hence of its components). Within the context of Computational Auditory Scene Analysis (CASA), a musical note is a three-dimensional auditory object (in energy, time and

frequency). In polyphonic music, events overlap both in the time and the frequency domain, meaning that transcription systems should be able to analyse the signal in both domains in order to return an accurate representation of the "scene". How the frequency-domain information is related across time highly influences the accuracy of the detection. This is specially important for real recordings, where the acoustic conditions greatly affect the behaviour of the signal.

5.4 An exploration into the grouping of information in the frequency-domain

Bearing all mentioned issues in mind, we decided to explore the detection of pitch in the frequency-domain. This will allow us to test all above-mentioned concepts and to propose a working-framework for a transcription system. The system is developed to deal with polyphonic real recordings of single instruments. Tests are specifically done in piano music. By considering the multi-dimensional nature of our analysis data, the system has been divided in two working blocks: a frame-by-frame analysis block, and a block for grouping information in time.

5.4.1 Frame by frame analysis

Spectral peak-picking

Let us consider $s(n)$, $n = 0 \dots N-1$, a N-length segment of the analysed polyphonic signal. By applying the FFT to the windowed segment (using a hanning window), the frequency-domain representation of $s(n)$, $S(k)$, $k = 0 \dots N-1$ is obtained (see Fig. 5.6(a)). The frequency resolution is $\Delta f = f_s/N$, where f_s is the sampling frequency.

The main observation behind all frequency-domain pitch estimation methods is that periodicities in the time-domain are represented as peaks in the frequency-domain. Hence, frequency-domain peak-picking is an obvious first step in our analysis process. Unfortunately, as can be observed in

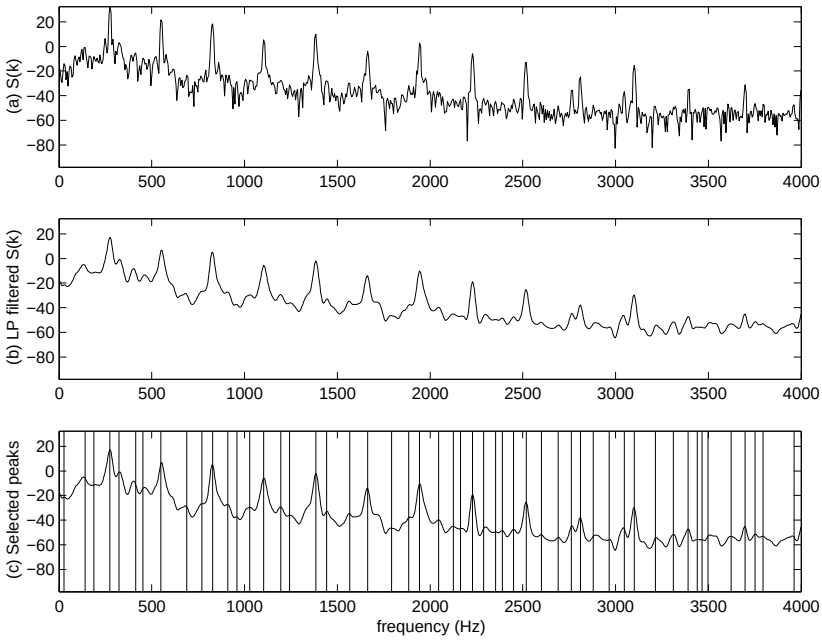

Figure 5.6: Spectral peak picking: original spectrum (a), low-pass filtered spectrum (b) and selected peaks (c).

Fig. 5.6(a), a spectrum profile is noticeably noisy and peak-picking on these data will translate in the inclusion of irrelevant information that may divert computational resources of the system and mislead the detection procedure. To avoid this, the spectrum is low-pass filtered using an infinite impulse response (IIR) digital filter with canonical:

$$y(n) = \sum_{i=0}^{M_a} a_i x(n-i) + \sum_{i=1}^{M_b} b_i y(n-i) \tag{5.2}$$

a and b, the filter coefficients, are estimated using the bilinear z-transform method for developing a Butterworth analogue prototype filter into a practical digital filter [MGT98]. The filtering procedure is as follows: first the input spectrum is filtered in the forward direction, then, the filtered spectrum is reversed and filtered again. The output of this second filtering

operation is reversed and used as the final filtered spectrum. This is done to avoid phase distortion. The result is shown in Fig. 5.6(b). The peaks that can be detected now correspond to the highest and more relevant peaks of the original spectrum. Once peak-picking is performed on the filtered spectrum (using a simple local maximum algorithm), peaks are matched to the closest local maximum of the unfiltered spectrum. This compensates for the loss of resolution that the filtering operation implies. The modified spectrum $S_p(k)$ of detected peaks (seen in Fig. 5.6(c)) is defined as:

$$S_p(k) \begin{cases} S(k) & \forall\, k \in [0, (N/2) - 1], \text{ where peaks are detected} \\ 0 & \text{elsewhere} \end{cases}$$

Designing a harmonic comb

Peak-picking by itself does not return any meaningful information about the notes being played during the length of the segment $s(n)$. Only by analysing magnitudes and positions of those peaks can significant conclusions be drawn.

As mentioned before, musical tones are expected to produce patterns in the frequency-domain with lobes approximately at mf_0, $m = 1 \ldots M$, where M is the number of lobes and f_0 is the fundamental frequency of the tone. These patterns are known as harmonic combs. Detection of such patterns is a three step process: first, estimation of the comb's root; second, generation of the target comb; and finally, estimation of the relation between comb and the information gathered at the peak-picking stage.

To determine the roots of the combs that better explain $S_p(k)$, we opted for the approach proposed by Lepain [Lep99]. In this method, it is assumed that $max\{S_p(k)\}$, the peak with the greatest magnitude of our spectrum, corresponds to one of the first Z partials of one or more tones present in the segment $s(n)$. Note that the use of this approach favours low-frequency hypothesis, as it does not necessarily require the presence of the fundamental to generate a possible note or hypothesis. The procedure is repeated for the P maximum peaks of the spectrum. This creates a $P \times Z$ matrix of

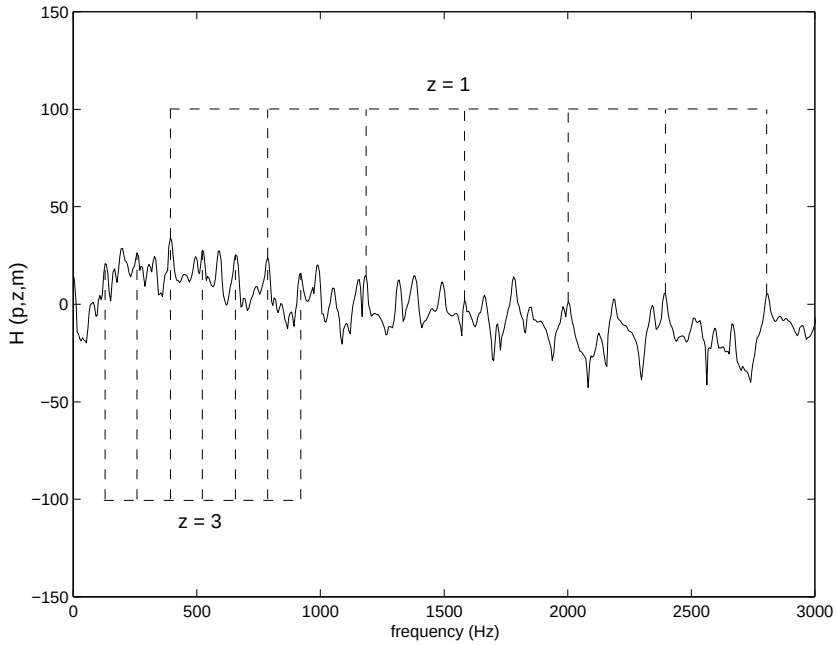

Figure 5.7: Harmonic combs for two different roots ($z = 1$ and $z = 3$) that include the maximum peak of $S(k)$.

fundamental frequency hypotheses defined as:

$$f_0(p, z) = \frac{f_i(k_p)}{z}, z = 1 \dots Z \tag{5.3}$$

such that:

$$k_p \equiv \text{bin of the } p^{th} \text{ maximum of } S_p(k), p = 1 \dots P \tag{5.4}$$

where $f_i(k)$, $k = 0 \dots (N/2) - 1$, is the bin instantaneous frequency whose calculation is explained in section 3.2.3. For each of the hypothetic $f_0(p, z)$, a target comb $C_{p,z}(k)$ is generated as:

$$C_{p,z}(k) = \begin{cases} 1 & \forall\, k \in k_m \equiv N\frac{f_0(p,z)}{f_s} m,\ m = 1 \dots M \\ 0 & \text{elsewhere} \end{cases}$$

where $M = min\{M_{max}, fs/2f_0(p,z)\}$. The target comb $C_{p,z}(k)$ is only a reference for a harmonic grid of an hypothetic f_0. It provides us with the expected location of the spectral information related to an assumed tone (peaks in the positions of the partials). It is expected that not all $P \times Z$ hypotheses are present in the segment $s(n)$, therefore it is necessary to evaluate how well the expected comb fits the data from our spectral peak-picking algorithm.

Let us define L_m, a group of bins of our spectrum such that their instantaneous frequencies $f_i(L_m) \in [f_m 2^{-1/24}, f_m 2^{1/24}]$ are within a quarter tone distance from $f_m = mf_0(p,z)$, the expected frequency of the m^{th} comb's lobe. The hypothesis associated with this comb, $H_{p,z}(m)$ can be defined as:

$$H_{p,z}(m) = max\{S_p(L_m)\},\ m = 1 \ldots M \tag{5.5}$$

The array $H_{p,z}(m)$ represents the closest approximate to each ideal comb $C_{p,z}(k)$ from the selected spectral data $S_p(k)$. Peaks are searched for in the vicinity of each comb's lobe. An example is shown in Fig. 5.7. For the maximum of $S_p(k)$, two harmonic grids are selected. The resulting arrays: $H_{1,1}(m)$ and $H_{1,3}(m)$, are shown for $M_{max} = 7$. In this example, spectral peaks are found in the spectrum that correspond to the expected positions of both combs. Note that this is not always the case as most combs will not find enough supporting information within $S_p(k)$.

Rating and selecting hypotheses

Initially, from the analysis of the magnitude of $S(k)$, $S_p(k)$, the matrix of selected spectral peaks, was created. Then the highest peaks of $S_p(k)$ were used to generate ideal harmonic grids, whose "real" approximates are stored in the array H. It is now expected that from the analysis of this array, hypotheses can be selected that correspond to the notes composing the polyphonic mixture in our segment $s(n)$.

H is organised in P groups of Z hypotheses each. Selection is performed through elimination of the weakest hypotheses, both through individual and competitive considerations. The process is performed at two levels: individually and competitively within groups, and competitively between group winners. The rules used for the analysis are explained as follow:

Individual rules

These rules are about the basic conditions for a hypothesis to be accepted without consideration of the context or "competition". They are absolute, a hypothesis rejected under these simple rules is not considered for further analysis. They both need to be satisfied. They are:

1. Minimum support: when constructing H, peaks are searched for in the vicinity of the expected position of a partial. If no spectral peaks are found within that region it is said that there is no support from that partial. For a hypothesis to be considered, a minimum amount of support (minimum number of detected partials) is required. This quantity is measured proportionally to the expected number of partials, avoiding bias according to frequency (for high-pitched hypotheses less support is always expected due to the limits imposed by the sampling frequency).

2. Minimum energy: the sum of the energy of all supporting partials (or total energy of the comb) must be above a minimum energy threshold. There are several factors (i.e. noise, reverberation, interactions between notes) that may produce undesired hypotheses. Although it is clear that this simple rule is not able to handle all of them, it is intended as a first-level filter that will save unnecessary processing.

Competitive rules

This is a slightly more complicated set of rules, that tries to include high-level knowledge into the decision-making procedure. As already stated, compe-

tition acts both between hypotheses in a group and between winners of all groups. It is worth noting that there is not a pre-specified number of winners per group or per frame. More than one winner could arise at any stage, or no winner at all.

The competition is based on being "explained" by the other hypotheses. In this context, "explained" means the justification of a hypothesis by the presence of other hypotheses rather than by its own presence. Comparisons are made in pairs (one hypothesis trying to explain another) or in groups (a hypothesis being explained from the interaction between two or more hypotheses). The winners are those that cannot be explained at all. The competitive rules are:

1. Detection of sub-harmonics: if there is an octave relationship between two hypotheses (their fundamental frequency ratio f_{high}/f_{low} is approximately an even number), the partials of the higher note are all overlapped by the partials of the lower note. A possible scenario is that the lower hypothesis corresponds to a sub-harmonic of the higher note, capitalising on the energy produced by the presence of the latter. When this is the case, some observations can be made. First, the even partials of a sub-harmonic will present an abnormal concentration of energy when compared with its odd partials. This can be expressed as:

$$\sum_{m \text{ odd}}^{M} H_{p,z}(m) \ll \sum_{m \text{ even}}^{M} H_{p,z}(m) \tag{5.6}$$

   Second, the energy of the first J partials (the very first of the comb) of the lower note will be notably smaller than the total energy of the note:

$$\sum_{m}^{J} H_{p,z}(m) \ll \sum_{m}^{M} H_{p,z}(m) \tag{5.7}$$

   If any of these observations is true for the lower hypothesis of an

"octave-related" pair, it is said that such hypothesis can be explained by the higher one.

2. Detection of overtones: another scenario for an octave-pair, is that where the higher hypothesis is an overtone of the lower. This is common source of mistakes in spectral-based approaches as mentioned during the bibliographical review. As all the information of the higher hypothesis is overlapped by that of the lower, we do not have enough "individual" data to decide about this note. Our only clue is the distribution of the energy through all the comb lobes. An abnormal concentration of the total energy in the first J partials of the highest note may be indicative of it just being the higher part of a lower tone. The observation may be stated as:

$$\sum_{m}^{J} H_{p,z}(m) \approx \sum_{m}^{M} H_{p,z}(m) \tag{5.8}$$

   If true, it is said that the lower hypothesis explains the higher one.

3. Harmonic overlapping: two notes in harmonic relation share partial components according to the relationship $af_{low} = bf_{high}$, where a and b are integer values. There are two situations that can be created by harmonicity and that we intend to detect. They are better explained with an example:

   The note G_5 (391.995 Hz) and the note C_5 (261.626 Hz) are harmonically related such that $3C_5 = 2G_5$. Therefore, every third partial of C_5 is overlapped by every second partial of G_5 and viceversa. Given the right conditions, an artificial G_5 could be identified due to the presence of C_5. In the same way, E_5 (329.628 Hz) is related to C_5, such that $5C_5 = 4E_5$. If a chord $E_5 + G_5$ is played, peaks will be generated approximately at the position of every third and fifth partial of C_5 which is not present (that is five of the first ten partials). This creates a hypothesis with strong support for C_5.

To avoid this, for all hypothesis in a competitive group the partial overlapping is calculated due to the presence and to the interaction between all other notes in the group. If the energy of all non-overlapped partials (those that cannot be explained by any of the other notes nor their interaction) is too small compared with the total energy of the comb, then it is said that the analysed hypothesis can be explained by the group of hypothesis. This is equivalent to:

$$\sum_{m}^{M} H_{p,z}(m) \gg \sum_{m \notin h}^{M} H_{p,z}(m),\ h \equiv \text{overlapped partials} \tag{5.9}$$

The rule is applied to all harmonically related hypotheses with the exception of octave intervals.

4. Common hypotheses: this a very simple rule applied to the "winners" group, that eliminates common hypotheses (hypotheses with f_0 mapping to the same note) coming from the first stage. It selects the better hypothesis, i.e. the one with the biggest and highest support.

5. Competitive energy: another simple rule, applied at the very end of the process. If there is still some competition after all rules are applied, the total energy of each of the remaining hypothesis is measured against the maximum energy of the group. Hypotheses with much lesser total energy than the overall maximum energy are rejected.

The complete process is illustrated in Fig. 5.8. Note that while individual rules only act at group level, most competitive rules (for sub-harmonics, overtones and harmonicities to be precise) act both at group and winners level. The simple rules of common hypotheses and competitive energy are designed exclusively for the winner's stage. "Surviving" hypotheses are the output of the frame-by-frame estimation system. They are henceforth named *frame hypotheses*, and denoted as $H_f(f_0, \tau_f)$, where τ_f is the corresponding frame index.

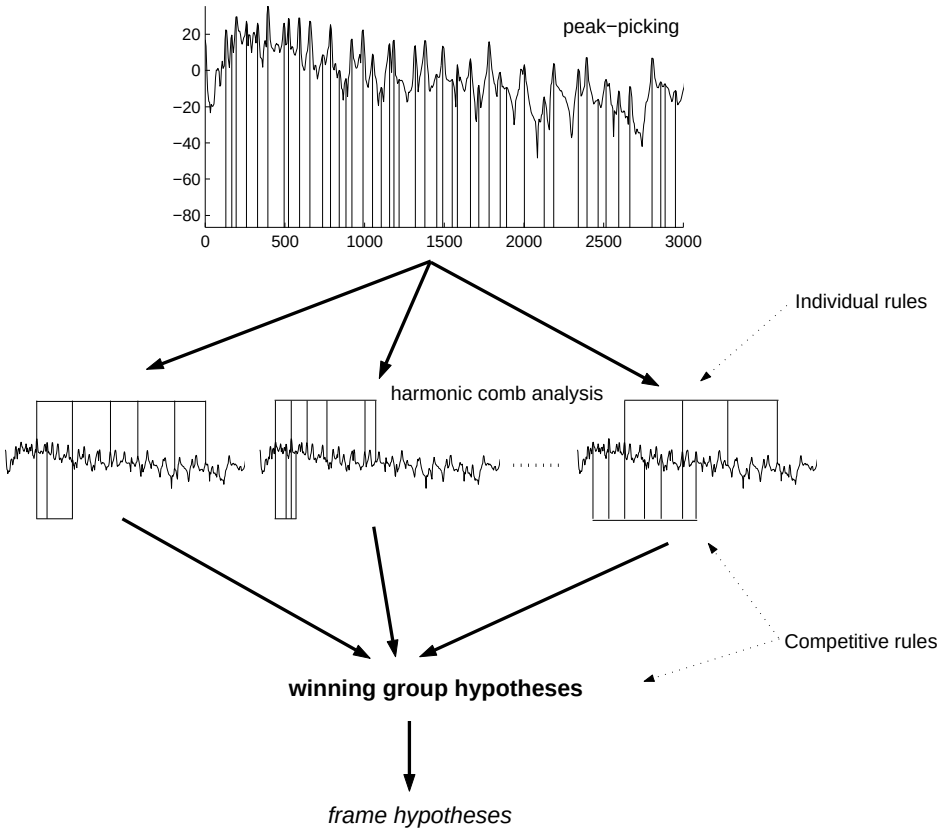

Figure 5.8: Block-diagram of the frame-by-frame pitch estimation process: spectral peak-picking (top) is followed by the design and evaluation of harmonic comb patterns. Individual and competitive rules are used for that evaluation.

5.4.2 Grouping information in time

Frame hypotheses are not completely reliable. The evolution of a sound is far from being an instantaneous phenomenon that can be captured in a "photographic" analysis. Even when a note is sustained, the corresponding pattern in the frequency-domain varies considerably, hence affecting the response of our frame-by-frame detection system. The situation is much more complicated when facing the time changes of a melodic contour within a polyphonic mixture. Consistency is difficult to maintain between frames. For all this,

frame-by-frame analysis needs to be complemented by the analysis of information along the time axis. The analysis explained in the following, intends to test the consistency and reliability of frame hypotheses and to evaluate their correspondence with the time-frequency representation of the signal. It is performed every signal's half second with information of the last second of analysis. The overlapping is necessary to avoid miss-judging information at the borders of the time window.

Duration analysis and gap filling

The first part of our analysis in time considers all frame hypotheses within the time-analysis window and evaluates their continuity and reliability based on two simple criteria: spacing between hypotheses and duration of closely-spaced groups of hypotheses. The intention is to eliminate spurious detections and to provide unambiguous support for continuous estimations.

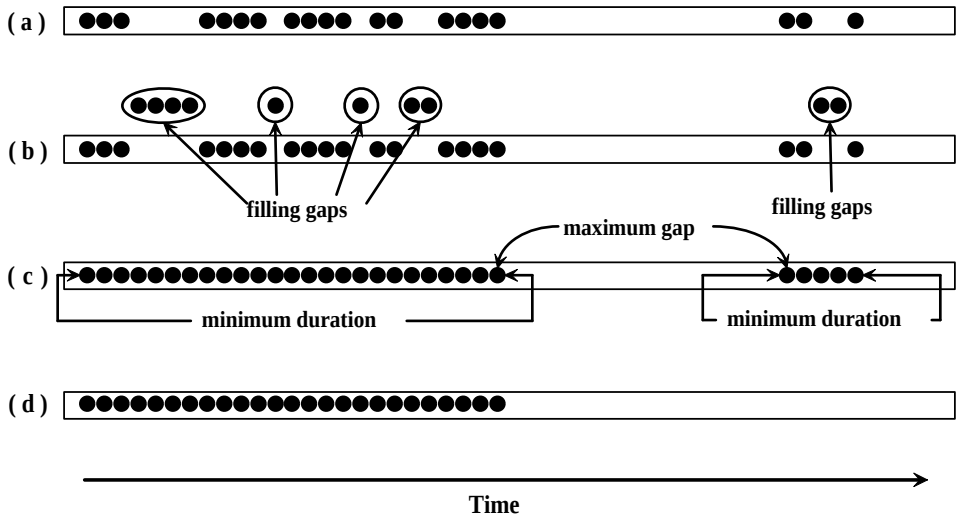

Figure 5.9: Continuity and duration analysis of a single strip of frame hypotheses:the original strip (a), small gaps being filled (b), separation of events and evaluation of their length (c) and the resulting strip (d).

To perform the analysis, the list of frame hypotheses is separated according to their f_0. Individual "strips" of hypotheses are analysed indepen-

dently. The direction of the analysis is forward (from beginning to end of the time-window).

We will use the example in Figure 5.9 to illustrate the process. The original strip (in Fig. 5.9(a)) presents several groups of consecutive frame hypotheses. As it stands it implies that several events of various durations are played within the time window. However, it can be noted that some gaps between events are small. It possibly means that these events are actually part of a larger event $H_e(f_0, \tau_i, \tau_e)$, where f_0 is the fundamental frequency of the analysis strip, τ_i is the initial frame of the event and τ_e is the ending frame of the event. For a frame hypothesis $H_f(f_0, \tau_f)$ this can be generalised as follows:

$$H_f(f_0, \tau_f) \in H_e(f_0, \tau_i, \tau_e), \tau_f \in [\tau_i, \tau_e] \text{ if } \nabla\tau_f < \delta_g \tag{5.10}$$

where $\nabla\tau_f$ is the difference between the position of two frame hypotheses, and δ_g is the maximum gap. Note that δ_g is a predetermined value. If the condition is true for non-consecutive frame hypotheses, then it is necessary to fill the gaps to avoid inconsistencies. This is illustrated in Fig. 5.9(b).

On the other hand, if $\nabla\tau_f \geq \delta_g$, then a new event is defined and the duration of the previous event is evaluated. Note that the duration d of an event is a function of its initial and final times. It is also evaluated against a predefined threshold δ_d, such that:

$$d(\tau_i, \tau_e) = \tau_e - \tau_i = \begin{cases} \geq \delta_d & \text{then } H_e(f_0, \tau_i, \tau_e) \text{ is kept} \\ < \delta_d & \text{then } H_e(f_0, \tau_i, \tau_e) \text{ is eliminated} \end{cases}$$

The evaluation of the duration is shown in Fig. 5.9(c). The final output of the analysis is in Fig. 5.9(d).

Following F0 energy profiles

The last stage of our time-analysis process, compares remaining event hypotheses against the energy profile of their fundamental frequencies along

the time axis. The idea is to assign an initial time location to detected events and to differentiate between a sustained note and repetitions of it. A one second buffer of the STFT of the signal is kept during the analysis.

The algorithm searches for all remaining f_0 within the current time window. For each event hypotheses, bins of the STFT block are selected such that they are within a quarter tone interval of the expected bin for f_0. Then, the energy profile E_0, for f_0 along the time axis is calculated as:

$$E_0(\tau) = \sum_k |S(k,\tau)|^2,\ k \in [|N\frac{f_0}{f_s}2^{(-1/24)}|, |N\frac{f_0}{f_s}2^{(1/24)}|] \tag{5.11}$$

To determine the location of energy increases within this particular frequency band, the derivative of the log energy is calculated (see chapter 4). This provides an initial guide for the time-positioning of detected events. Note that this alignment of events is independent for each frequency band. This algorithm does not compare onset information across frequencies. That would be part of the integration of the multi-pitch estimation systems with the developed onset detection system.

Relevant peaks (those about a certain threshold) are detected on the derivative profile and their position compared with the starting time of detected events within the same frequency strip. There are five cases considered as shown in Fig. 5.10:

1. *Observation:* There is no interference between an event starting time and its preceding energy peak (Fig. 5.10(a)). This is the most common situation. The detection is expected to be more robust during the steady-state of a note and prone to failure during transients, therefore a delay is expected between the onset time and the beginning of the detected event. *Action:* To change the starting time of the event hypotheses to that of the immediately preceding energy increase. Even with long delays (inside our time-window), this offers the best explanation for both observations.

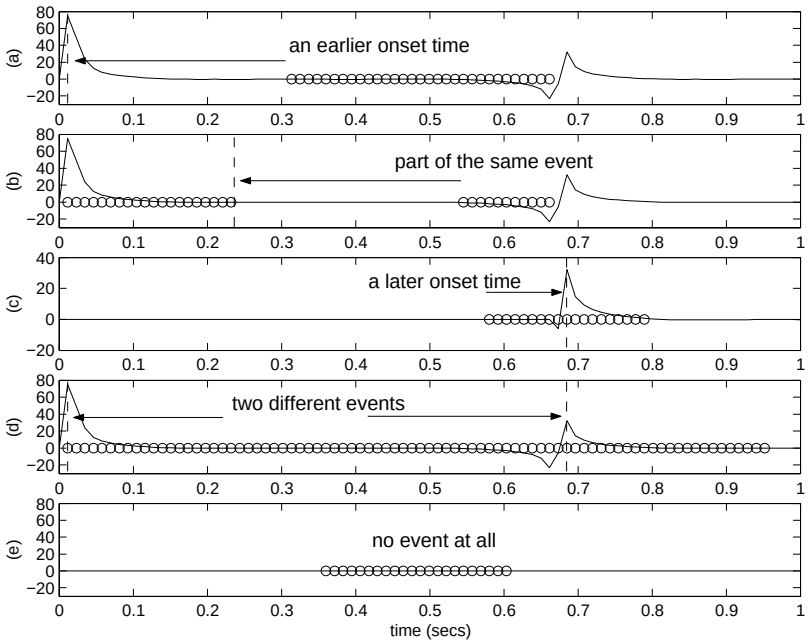

Figure 5.10: Comparison between f_0 energy profile and detected events.

2. *Observation:* The ending of the previous event is closer than the preceding energy peak (see Fig. 5.10(b)). This is the case when the detection is being interrupted due to temporal changes in the sound or to the action of additional sounds (i.e. other notes, noise, etc). *Action:* To integrate the two events, creating a single event with the starting time of the first and the ending time of the second.

3. *Observation:* An event with no preceding but a succeeding onset (see Fig. 5.10(c)). This is a rare observation, possibly due to the presence of long sustained notes or reverberation. *Action:* To modify the starting time of the event to that of its succeeding energy increase, but without modifying its stopping time. If the resulting duration is not enough to maintain the hypothesis according to the "duration analysis" rules, then the whole event is simply eliminated.

4. *Observation:* More than one onset detected during the length of the

event (see Fig. 5.10(d)). Repeated notes are a common problem in transcription systems. The detection is continuous for all involved frames but there is more than one event actually occurring. *Action:* To separate the event into multiple events with starting times corresponding to the detected onsets and durations sufficient to allow the detection according to the above-mentioned rules.

5. *Observation:* No energy increases can be associated to an event (see Fig. 5.10(e)). *Action:* This is only expected for lower frequencies, when the fundamental frequency is not necessarily present or when facing certain musical features, i.e. glissando or legato. The action, adjusted to piano experiments, is to eliminate the event if its frequency is higher than a certain threshold.

5.4.3 Integration into the blackboard framework

As mentioned, the current system is divided in two working blocks, for frame-by-frame (Fig. 5.11) and temporal (Fig. 5.12) analysis. Note that due to the complexity of the system the scheduler has been omitted from the drawings, however the performance of the system remains consistent with the main principle of the scheduler's operation: to generate the highest possible level of information given the current status of the blackboard.

The frame-by-frame block is depicted in Fig. 5.11. Its operation is related to four levels of the original blackboard database as presented in Chapter 2: those of the spectral magnitudes and instant frequencies provided by the phase-vocoder analysis, the group hypotheses and the frame hypotheses levels.

The information regarding spectral magnitudes and instant frequencies is used to generate a list of peaks (amplitude maxima with their corresponding frequency values) in the spectrum, through the bottom-up operation of **KS_spectralpeaks**. These peaks are stored in the bottom sub-level of the internal hierarchy corresponding to the *group hypotheses*. This information is used by the bottom-up **KS_roots** to generate its namesake sub-level of

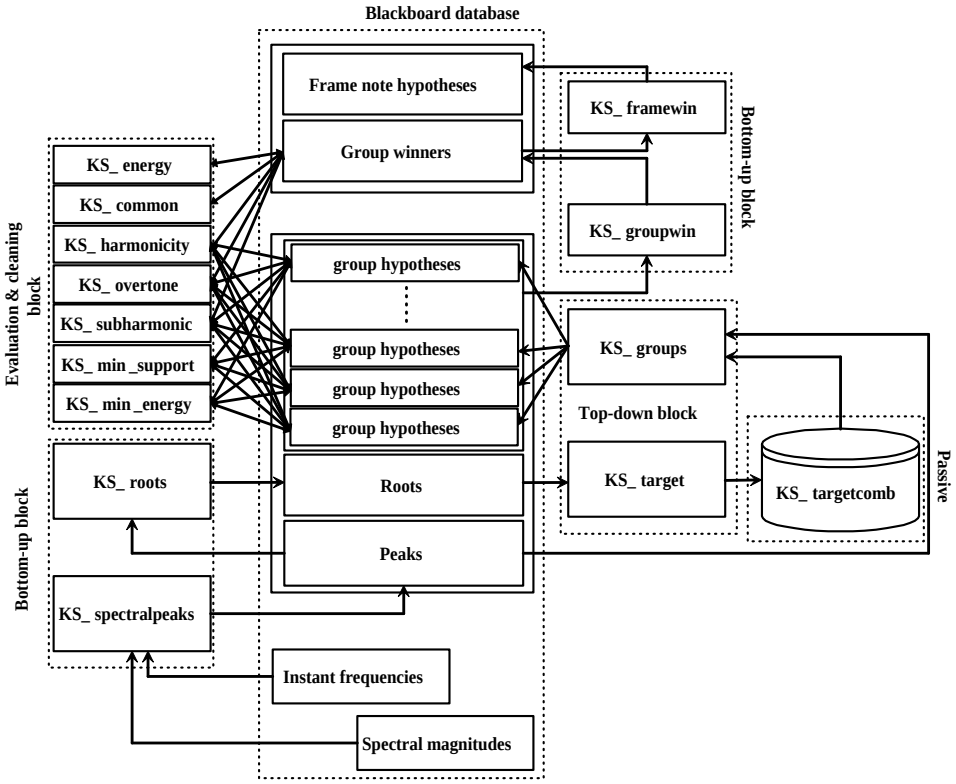

Figure 5.11: Integration of the frequency-domain pitch detection system into the blackboard framework: the frame-by-frame working block.

the hierarchy (according to the method previously explained).

These roots are used to generate the information to be stored in the passive knowledge source **KS_targetcomb** by the top-down **KS_target**, and according to the high-level knowledge related to the expected positions of partials (harmonic grids). Subsequently, spectral peaks information and target combs are used by **KS_groups** to generate the P groups of hypotheses that start our frame competition.

Both individual and competitive rules are implemented as *Evaluation and cleaning* knowledge sources in the practical implementation. They evaluate hypotheses according to their specific knowledge and their ratings are stored in the database as part of the hypotheses information. Individual rules (the ones contained in **KS_min_energy** and **KS_min_support**) are

applied only to group hypotheses, while most competitive rules (except those applied only to *group winners*: **KS_common** and **KS_energy**) are applied to both group and frame hypotheses.

After the rating has been performed, bottom-up knowledge sources are used to select high-rated hypotheses and take them to the next level. The first of such knowledge sources **KS_groupwin** feeds the *group winners* sub-level of the hierarchy with the fittest group hypotheses. Similarly, **KS_framewin** constructs a final list of *frame note hypotheses* with the winners of the competition at the previous level. As mentioned before, the "competition" is based on the explanation of a hypothesis as a result of the interaction between the other hypotheses. The less "explained" a hypothesis is, the better the chance of survival.

The second working block is the implementation of the temporal analysis as explained in section 5.4.2. It is depicted in Fig. 5.12.

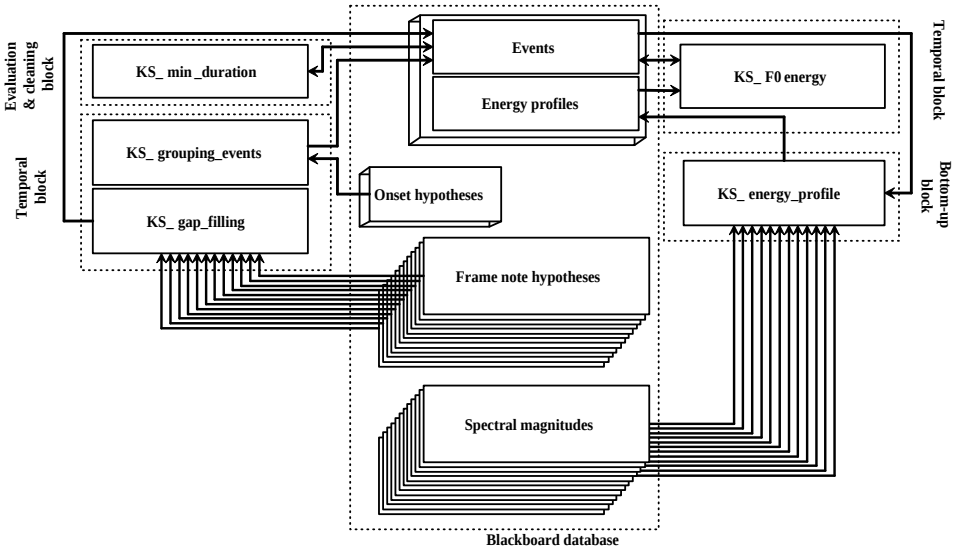

Figure 5.12: Integration of the frequency-domain pitch detection system into the blackboard framework: the temporal working block.

As with the calculation of onsets, a temporal window is used for the analysis of detected pitches and their grouping into note events. This is

done, once all frame calculations are completed within the window. A temporal knowledge source, **KS_gap_filling**, is used to analyse continuity of the estimated hypotheses. Hypotheses are grouped into individual strips (according to their pitch). Small gaps between detections are filled according to the procedure previously explained. The results of this operation are stored in the *events* level of the hierarchy. This is the highest level in the blackboard and what remains there, after the analysis is performed, is used as the output of the system.

Three knowledge sources operate on the events' information. First, the *evaluation and cleaning* **KS_min_duration** evaluates that proposed events are present for a minimum amount of frames. This consistency is necessary for a note to be recognised. Events that do not comply with this condition are erased from the database. Then, the energy profile of the fundamental frequency of the analysed note is considered within the temporal window. The profile is generated by a bottom-up knowledge source, **KS_energy_profile**, from the analysis of the corresponding frame spectral magnitudes. **KS_F0energy**, a *temporal* module, uses these energy profiles and implements the *observations* and *actions* logic explained in the previous section to eliminate or modify current events. Finally, the also *temporal* **KS_grouping_events** uses the information from the onset analysis to group note events into chords, and to verify coherence between both onset and note information. The starting times of notes are modified according to detected onset times, and notes with no onset support are eliminated.

5.4.4 Results and discussion

As with previous similar work, we suffer from the lack of a standard for evaluating pitch estimation systems. Perception is not easy to quantify, especially in expert tasks such as this. Several approaches to the system's evaluation were considered (e.g. comparing against human transcriptions of the same music, evaluating perceptual similarity on test audiences, etc.), however, a straightforward comparison between results and target scores

was favoured as a means to address the need for quantitative results.

However, providing a target score for real recordings is not exempt of complexities. Available scores for recorded music never correspond exactly to what is being performed in the recording, as musicians change tempos, make mistakes, add ornaments, etc., all of which affect the operation of the system. The approach taken for the onset detection algorithm's evaluation, hand-labelling of events in real recordings, requires huge musical expertise in the case of polyphonic pitch estimation and is more susceptible to errors.

Given all this, we opted for the creation of real recordings from known scores of piano music. MIDI files generated from the live performance of amateur and professional players were used on a MIDI-controlled acoustic grand piano. This allowed us to compile a small database of academic piano music on which to evaluate the performance of our approach to music transcription. Recordings were made at 44100 Hz sampling rate, in stereo by using a coupled pair of condenser microphones in a purpose-built studio room. The system's input data consists of PCM mono wave files at 22050 Hz sampling rate. The length of the analysis window is 200 ms and overlapping frames are separated by a 10 ms hop.

There are 4258 notes in the used test-bed corresponding to segments of piano pieces by five well-known composers: Wolfgang Amadeus Mozart, Ludwig van Beethoven, Claude Debussy, Scott Joplin and Maurice Ravel. The selection of composers and musical pieces was arbitrary.

An example of the transcription results can be seen in Figure 5.13. The spectrogram of the analysed audio file is in 5.13(a), 5.13(b) shows the original MIDI file and 5.13(c) the estimated MIDI. The estimated MIDI file is rendered from the detections of the system. The correspondance between MIDI note numbers and their fundamental frequency in Hertz is given by:

$$MIDI_{f_0} = \|69 + 12 \cdot \log_2(\frac{f_0}{440})\| \tag{5.12}$$

It can be noticed that note durations differ between the original and the estimated MIDI files. This is important, because MIDI files contain only

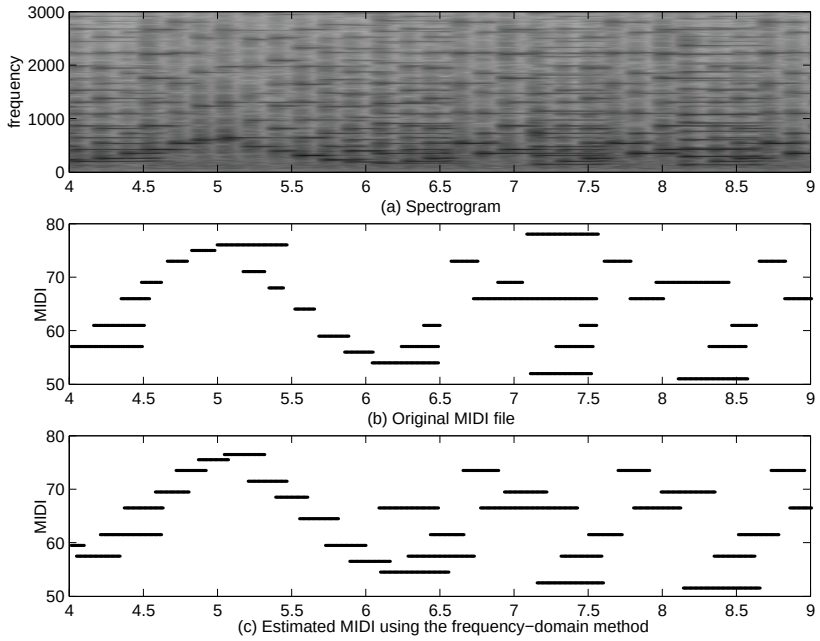

Figure 5.13: Polyphonic pitch estimation on a segment of a Debussy's piano piece: (a) spectrogram, (b) original MIDI file and (c) estimated MIDI file

control commands that are "blind" to the characteristics of the instrument being played. This means that, even when a note-off command is being produced by the controller, the acoustic piano may produce the corresponding sound for a few more moments. Moreover, the MIDI does not consider the room acoustics (e.g. reverberation), the effect of which is captured on the audio recordings. The combination of these factors may produce, for example, higher polyphonies that the ones nominally expected.

Apart from these subtle differences, there are the obvious mistakes produced by the limitations of the estimation system. An algorithm was developed to compare both MIDI files and to quantify the "correctness" of results. Table 5.1 shows results by composer and overall results.

Numbers on the table correspond to the rate of good detections (GD): original notes correctly detected; the rate of false negatives (FN): original

Composer	% GD	% FN	% RECALL	% FP
Mozart	72.78	27.22	85.47	14.53
Beethoven	65.99	34.01	84.58	15.42
Debussy	74.05	25.95	84.10	15.90
Joplin	60.77	39.23	81.41	18.59
Ravel	68.83	31.17	72.60	27.40
TOTAL	68.98	31.02	83.60	16.40

Table 5.1: Note estimation results using the frequency-domain approach

Composer	Average polyphony	Max. polyphony
Mozart	1.7104	4
Beethoven	2.0875	8
Debussy	1.9932	7
Joplin	3.5590	7
Ravel	3.6664	8

Table 5.2: Average and maximum polyphony for test files

notes not detected; the rate of recall (RECALL): estimated notes that are correct; and the rate of false positives (FP): estimated notes that are not present in the original.

The overall rate of good detections is almost 70% of the total amount of notes. The best case, that of Debussy's piano pieces reaches 74% percent of good detections while the worst case (Scott Joplin's segments) is slightly over 60%. This is interesting if we consider that while Debussy's segments are based on chromatic melodic progressions, usually with low polyphonies involved, Joplin's segments are rich in complex polyphonies of highly harmonic sounds (as rag-time music usually is). Table 5.2 illustrates average and maximum polyphonies by composer. It is worth insisting that these are nominal polyphonies, probably lower than those in the audio files.

There is a slight correspondence between the quality of the results and

the polyphony of the analysed pieces. However, as suggested before, the size of the polyphony is only part of the problem. The quality of the polyphony is of major importance in these results: the increase on the appearance of highly harmonic intervals is expected to increase the number of errors. Also the presence of thrills and repeated notes is a common source for miss-detections. Table 5.3 categorises the false negatives per composer into the most common sources for errors.

Categories on the table correspond to false negatives produced by octave intervals, "higher" when the higher note of the interval is missing or "lower" when the lower note of the interval is missing; thirds, when the missing note is the third of another note; fifths, when the missing note is the fifth of a present note; and repeated notes, when the missing note is preceded in a short temporal window by another note of the same pitch. Rates in the table are over the totality of notes for each composer. In total, these categories account for almost 50% of all false negatives.

Unsurprisingly, the higher note of a harmonic interval (octaves, thirds or fifths) is consistently the biggest cause for false negatives. *Octave (higher)* is the overall major false negative generator, being particularly critical in the case of Scott Joplin's segments. Debussy's pieces show the lowest rate for this error. This is consistent with our previous comment, the combination of high polyphonies with harmonically related intervals boosts the proliferation of errors. Thirds and fifths are usually high as well, emphasising the fact that harmonic overlapping is critical for the estimation of multiple pitches in real recordings.

The lowest note in an octave interval is not a high source of errors. This can be expected due to the method implemented on this system for the generation of the roots of the harmonic combs. As mentioned before, the assumption that the high peaks in the spectrum are part of the first few partials of a note, favours the estimation of lower notes in harmonic intervals. The balance of errors would be other than this if we had assumed the higher peaks to correspond to fundamental frequencies of candidate notes. Notably,

MOZART		BEETHOVEN	
False negatives = 27.22 %		False negatives = 34.01 %	
Octave (higher)	3.54 %	Octave (higher)	6.21 %
Octave (lower)	0.45 %	Octave (lower)	0.62 %
Thirds	4.60 %	Thirds	5.05 %
Fifths	5.28 %	Fifths	2.10 %
Repeated notes	0.38 %	Repeated notes	3.88 %

DEBUSSY		JOPLIN	
False negatives = 25.95 %		False negatives = 39.23 %	
Octave (higher)	3.43 %	Octave (higher)	11.00 %
Octave (lower)	0.76 %	Octave (lower)	8.61 %
Thirds	3.69 %	Thirds	0.32 %
Fifths	3.94 %	Fifths	3.51 %
Repeated notes	0.38 %	Repeated notes	0.00 %

RAVEL		**TOTAL**	
False negatives = 31.17 %		False negatives = 31.02 %	
Octave (higher)	4.76 %	Octave (higher)	5.50 %
Octave (lower)	2.60 %	Octave (lower)	1.90 %
Thirds	0.43 %	Thirds	2.28 %
Fifths	3.46 %	Fifths	3.71 %
Repeated notes	0.00 %	Repeated notes	1.36 %

Table 5.3: Frequency-domain method: categorisation of false negatives per composer

Joplin's results are the only exception. There, the lower notes of the octave interval highly contribute to the overall count of errors. This suggests that the chords used do not only span one but multiple octave intervals (in which case, the lower note of one interval is also the higher of another octave interval).

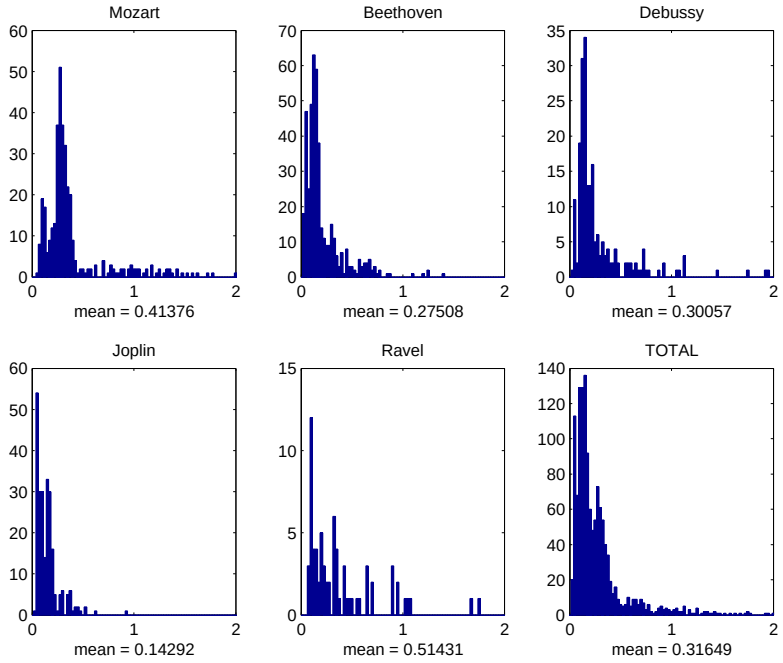

Figure 5.14: Distribution of false negatives' durations per composer. Means at the bottom of each plot correspond to the average duration of notes for that composer.

Although repeated notes' errors are consistently low in most files (with the exception of Beethoven's music), they are tightly associated with another factor that contributes to the proliferation of false negatives: the short duration of notes. In Fig. 5.14, the distribution of the durations of false negatives is shown. Means at the bottom of each plot correspond to the mean duration of the notes of the analysed music files. It can be noticed that false negatives are usually short lived notes. Their high concentrations are often below the

average duration of notes for each composer. Consistency of the detection is an important condition for the estimations performed by the system. It is extremely important in avoiding the proliferation of spurious detections by the system, but it also operates against high detection rates when the speed of the music increases and the duration of notes shorten. The shorter a musical note is, the more unlikely is that it can be consistently detected by the system.

Missing notes are not the only type of errors in the current analysis. Let us return to Table 5.1 and concentrate on the rate of recall for the analysed music files and on its counterpart, the rate of false positives.

The total recall rate (83.60%) is very high using the current method. It indicates that there is a high probability that estimated notes are present on the original file. However, particular cases, such as Ravel's tests, show that there is a difficult trade-off between over and under estimation. Previous results suggest a relaxation of the constraints imposed by the system while considering harmonic intervals. However, and as can be seen in Table 5.4, the major causes of false positives are not far from those that produce false negatives. Categories on the table correspond to mistakenly detected first harmonics, sub-harmonics, thirds, fifths and repeated notes. It can be seen that octave related errors are consistently the major source of false positives, reaching their highest level at Ravel's estimations. Fifths follow as important contributors, while thirds and repeated notes fall behind on their importance.

In total, these categories account for nearly 70% of all false positives, well beyond the contribution of their equivalents on the false negative case. This suggests that relaxation of constraints regarding these conditions will increase the amount of false detections significantly, while the contribution to the good detections rate will be less sharp. In fact, the system has been tuned to maximise results for piano files, and the developed set of parameters is already producing "optimal" results. Pushing the boundaries will only deteriorate the system's performance.

MOZART		BEETHOVEN	
False positives = 14.53 %		False positives = 15.42 %	
1^{st} harmonics	4.34 %	1^{st} harmonics	4.48 %
Sub-harmonics	2.04 %	Sub-harmonics	4.18 %
Thirds	0.97 %	Thirds	0.60 %
Fifths	1.42 %	Fifths	1.69 %
Repeated notes	0.53 %	Repeated notes	0.20 %
DEBUSSY		JOPLIN	
False positives = 15.90 %		False positives = 18.59 %	
1^{st} harmonics	2.89 %	1^{st} harmonics	3.21 %
Sub-harmonics	4.19 %	Sub-harmonics	4.06 %
Thirds	0.87 %	Thirds	1.28 %
Fifths	2.60 %	Fifths	1.92 %
Repeated notes	0.72 %	Repeated notes	0.43 %
RAVEL		**TOTAL**	
False positives = 27.40 %		False positives = 16.40 %	
1^{st} harmonics	5.94 %	1^{st} harmonics	4.04 %
Sub-harmonics	6.85 %	Sub-harmonics	3.64 %
Thirds	0.00 %	Thirds	0.83 %
Fifths	4.57 %	Fifths	1.99 %
Repeated notes	1.83 %	Repeated notes	0.54 %

Table 5.4: Frequency-domain method: categorisation of false positives per composer

An alternative evaluation

As mentioned in the introduction, the work explained in this dissertation is part of a bigger project, intended to develop a system able to search written music databases from real polyphonic music recordings. This research falls into the field of music information retrieval.

The interest lies in the fact that the object of the search and the query material are not in the same domain (symbolic versus audio). This creates the need for a comparison between indexed and query data in a middle level (that is between audio and score). The information provided by this system falls into this category as it is a high-level representation of the audio, not complete enough to be considered a score. However, it provides information on an abstract level, that of music. This is not possible with music recognisers currently available on the market, where an audio blueprint is developed and then compared against an audio database [AHH+01, NMB01]. For those systems there is no similarity between different recordings of the same musical piece.

Given all this, retrieval on a large score database based on the information produced by this system is an interesting option for the evaluation of its performance. Successful retrieval indicates that the estimation is capturing enough information about the musical content of the signal as to differentiate it from other musical pieces that may be similar (e.g. pieces from the same author).

Appendix A is a re-print of a paper where relevant theory and results of these experiments are included. We believe results are not necessarily related to the quality of the transcription, but to the perception of the similarity between original files and estimations (note that some of the tested files are from real CD recordings, while the corresponding scores of the database are generic scores for those pieces).

It is also an example of a practical application for systems such as the ones presented here.

5.5 Time-domain note identification

The previous system gave us a clear understanding of the problem in hand. It seems that without a radical paradigm shift, the underlying complexities of the process of polyphonic pitch estimation cannot be solved. It is also apparent that, without enhancing our knowledge about the signal, little can be achieved. We aim to propose an alternative approach that avoids, at least partially, the usual paradigm of analysis in the frequency domain. Also, we intend to propose a system that extracts knowledge about the signal, from the signal itself. The system, proposed in collaboration with Daudet [BDS02], uses a hybrid method, where the classical frequency-domain approach is improved by a time-domain recognition process. This enables a refinement of our results by taking into account the information contained in phase relationships, that are lost when only the magnitude spectra of sounds are analysed.

5.5.1 Linear additive approach

Let us return to $s(n)$, $n = 1 \dots N$, a segment of our analysis signal. Let x_i, $i = 1 \dots M$, be the time-domain normalised waveform of one of the individual notes of a single instrument. There is a group of assumptions associated to these x_i: Let us assume (as an initial approximation) that each x_i is independent of its loudness, that is, the waveform remains the same regardless of the strength at which the corresponding note has been played, except for a global scaling of the signal's amplitude.

Let us also assume that within a mixture a waveform is independent of the presence of other waveforms. Furthermore, let us demonstrate that the individual notes x_i form a family of linearly independent vectors. This means that it is not possible to obtain a note by a linear combination of other notes, as shown by contradiction: Let us assume linear dependency between the individual notes x_i, such that $\sum_i a_i x_i = 0$, for a_i not all equal to zero. Let i_0 be the smallest index, if any, such that $a_{i_0} \neq 0$. The signal

$x_{i_0}(n)$ contains the fundamental frequency corresponding to the i_0^{th} note, which is not present in any of the other x_i, $i > i_0$, therefore it is necessary that $a_{i_0} = 0$, hence the contradiction.

Let $\mathcal{D} = \{x_i\}_{i=1\ldots M}$ be the database containing the M waveforms of the individual notes. In this context, $s(n)$ can be simply defined as a weighted linear sum of the individual notes x_i, or:

$$s(n) = \sum_{i=1}^{M} \alpha_i \, x_i(n) \tag{5.13}$$

where α_i is the mixing coefficient for the i^{th} note, such that:

$$\alpha_i \begin{cases} > 0 & \text{if the } i^{th} \text{ note is played in } s(n) \\ = 0 & \text{otherwise} \end{cases}$$

α_i increasingly maps the loudness, or velocity, of the corresponding note.

Following this definition, the frame-by-frame polyphonic pitch estimation problem, can be re-stated as the calculation of the values of the mixing vector $\boldsymbol{\alpha} = \{\alpha_i\}_{i=1\ldots M}$, given the segment $s(n)$ and the database $\mathcal{D}$. This operation returns information about what notes have been played in the frame and with what velocity.

In finite dimension, a simple algebraic solution can be found to this problem. Let $\langle \cdot , \cdot \rangle$ be the canonical scalar product on $\mathbb{R}^N$, the space of real sequences of finite length N. Let x_j, $j \in [1, M]$, be a given vector of the database $\mathcal{D}$. By calculating the left scalar product of the segment with the given vector, we obtain:

$$\begin{aligned} \langle x_j, s \rangle &= \sum_{i=1}^{M} \alpha_i \, \langle x_j, x_i \rangle & (5.14) \\ &= \alpha_1 \langle x_j, x_1 \rangle + \alpha_2 \langle x_j, x_2 \rangle + \cdots + \alpha_M \langle x_j, x_M \rangle & (5.15) \end{aligned}$$

When expanding this to all $j = 1 \ldots M$, the following set of linear equations is obtained:

$$
\begin{aligned}
\langle x_1, s\rangle &= \alpha_1\langle x_1, x_1\rangle + \alpha_2\langle x_1, x_2\rangle + \cdots + \alpha_M\langle x_1, x_M\rangle \\
\langle x_2, s\rangle &= \alpha_1\langle x_2, x_1\rangle + \alpha_2\langle x_2, x_2\rangle + \cdots + \alpha_M\langle x_2, x_M\rangle \\
&\vdots \qquad\qquad\qquad\qquad \vdots \\
\langle x_M, s\rangle &= \alpha_1\langle x_M, x_1\rangle + \alpha_2\langle x_M, x_2\rangle + \cdots + \alpha_M\langle x_M, x_M\rangle
\end{aligned}
$$

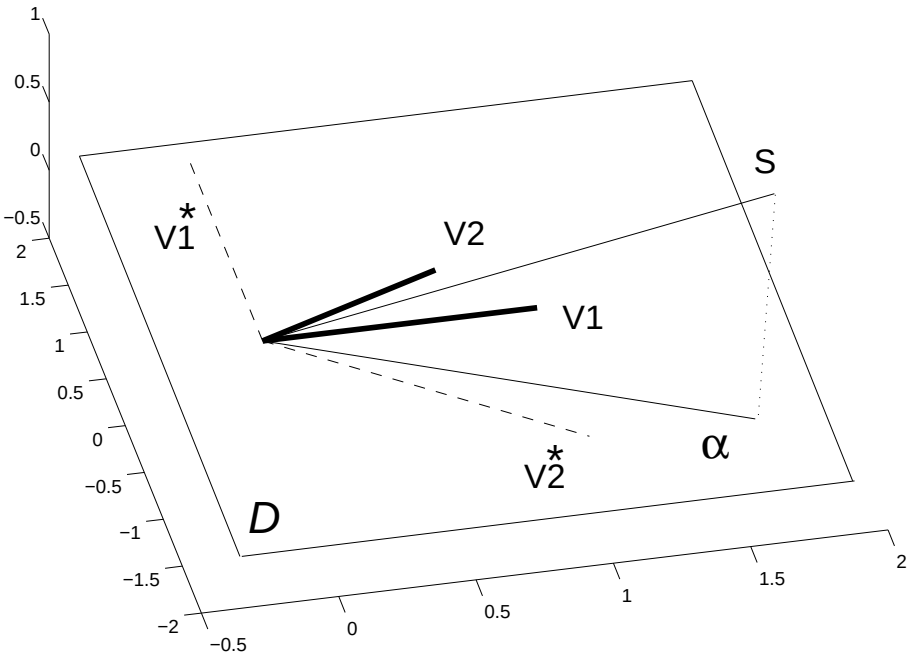

Figure 5.15: $\boldsymbol{\alpha}$ is the orthogonal projection of the vector s on the subspace $\mathcal{D} = \mathrm{Span}(v_1, v_2)$. Its components (α_1, α_2) in the basis $\{v_1, v_2\}$ are given by the scalar products with the dual basis $\{v_1^\star, \tilde{v}_2^\star\}$.

Let us also define D, the representation of $\mathcal{D}$ on $\mathbb{R}^N$, a $M \times N$ matrix whose rows are the N-length individual vector notes x_i:

$$
D = \begin{bmatrix}
x_1(1) & x_1(2) & \dots & x_1(N) \\
x_2(1) & x_2(2) & \dots & x_2(N) \\
\vdots & \vdots & \vdots & \vdots \\
x_M(1) & x_M(2) & \dots & x_M(N)
\end{bmatrix}
$$

Henceforth, the set of linear equations (equivalent to Equation 5.15) can be expressed in terms of D as:

$$D\ s = DD^T\ \boldsymbol{\alpha} \tag{5.16}$$

Because of the linear independence of the rows of D (as previously demonstrated), the $M \times M$ matrix DD^T is not singular, thus invertible. Hence, Equation 5.16 is equivalent to:

$$\boldsymbol{\alpha} = (DD^T)^{-1}\ D\ s \tag{5.17}$$

Therefore, the mixing vector α can be reconstructed by a simple matrix product between the fixed matrix $(DD^T)^{-1}D$ and the segment s. The rows of $(DD^T)^{-1}D$ form the dual basis $\mathcal{D}^\star$ of the basis $\mathcal{D}$; and $\boldsymbol{\alpha}$, which represents the orthogonal projection of s on the subspace $\mathcal{D}$, is obtained by scalar products with elements of $\mathcal{D}^\star$. This is illustrated in Figure 5.15.

5.5.2 Phase alignment

The main advantage of working with time-domain waveforms, is that the phase of the signal is taken into account. It is also the main disadvantage. Under certain specific conditions (i.e. audio files generated with synthesised sounds and "perfect" - quantised - playing), phase-alignment between simultaneous notes could be a possibility. However this is never the case when working with real recordings. Individual notes within a chord are never perfectly synchronous. Thus, the above-mentioned approach is over-simplified (as all scalar products assume alignment), and the results obtained using this method are not accurate.

Let us define τ_t, the shift-by-t-samples operator, such that:

$$\tau_t x_i(n) = x_i(n-t) \tag{5.18}$$

A possible solution for our phase-alignment problem is to consider all shifted versions of vectors x_i up to a delay T, or $\tau_t x_i$, $t = 1 \ldots T$. This

means that we will work in subspaces of higher dimension (such as $M \times T$).

Needless to say, the computational costs will dramatically increase. Besides, when $M \times T$ gets larger than N (as will happen for large values of T) the family of vectors becomes overcomplete, hence it is no longer linearly independent. This implies the non-invertibility of matrix DD^T , thus the vector $\boldsymbol{\alpha}$ cannot be calculated as stated above. An alternative approach to alignment is needed.

On a frame-by-frame basis, let us suggest that for the i^{th} note in the database, we only need to use the $\tau_{t_i} x_i$, such that t_i compensates for the phase misalignment between the sound $s(n)$ and the note x_i. This is equivalent to generalise Eq. 5.13 as:

$$s(n) = \sum_{i=1}^{M} \alpha_i \; x_i(n - t_i) \tag{5.19}$$

The particular delay t_i will be computed as:

$$t_i = \arg \max_{t=1...T} \langle \tau_t x_i, s \rangle, \; \forall i \in [1, M] \tag{5.20}$$

This approach, although slightly suboptimal when compared to the first proposed solution, is much easier to implement in practical terms. If considering all possible delays within the length of x_i ($T = N$), the scalar product $\langle \tau_t x_i, s \rangle$ becomes equivalent to the convolution of $s(n)$ and $x_i(n)$:

$$\langle \tau_t x_i, s \rangle = x_i * s \tag{5.21}$$

then Eq. 5.20 can be re-written as:

$$t_i = \arg \; \max\{x_i * s\} \tag{5.22}$$

We can now define an "aligned" database $\tilde{\mathcal{D}} = \{\tau_{t_i} x_i\}_{i=1...M}$ and adopt the procedure described in (Eq. 5.17) with the modified basis:

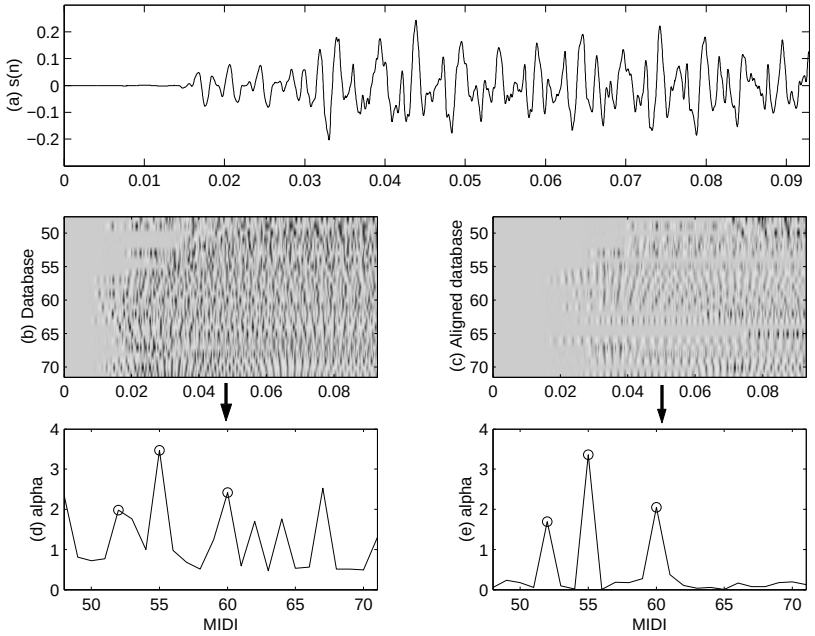

Figure 5.16: $\boldsymbol{\alpha}$ estimation on a segment $s(n)$ (a), using D (b) and $\tilde{D}$ (c). Estimation is more reliable with alignment (e) than without (d).

$$\boldsymbol{\alpha} = (\tilde{D}\tilde{D}^T)^{-1}\ \tilde{D}\ s \tag{5.23}$$

In Figure 5.16(a) a segment $s(n)$, containing the notes E4 - G4 - C5 (MIDI numbers 52 - 55 - 60), is shown. A database D and an aligned database $\tilde{D}$ (Fig. 5.16(b) and (c) respectively) are used for the note estimation procedure. It can be seen that the $\boldsymbol{\alpha}$ estimated using $\tilde{D}$ (Fig. 5.16(e)) provides accurate information about the notes being present, while its non-aligned counterpart (in Fig. 5.16(d)) presents some erroneous detections.

5.5.3 Results with a fixed database

To validate the above method, we have tested our recognition process on synthesised chords, made as weighted sums of known waveforms, with small random time shifts. The used waveforms belongs to the McGill University's

catalogue of individual sounds of orchestral instruments [OW89]. Specifically, we used recordings of the 88 individual notes of three different acoustic pianos (that we will conveniently term pianos 1, 2 and 3).

Experiments were made constructing the database D with the individual notes of *Piano 1*. When tested with chords produced with samples from the same piano, results were excellent, as even complex chords (eg. containing more than 6 notes with harmonic relations) could be correctly identified with great robustness. An example is shown in Fig. 5.17(a), where a C major chord is successfully identified using the linear additive method.

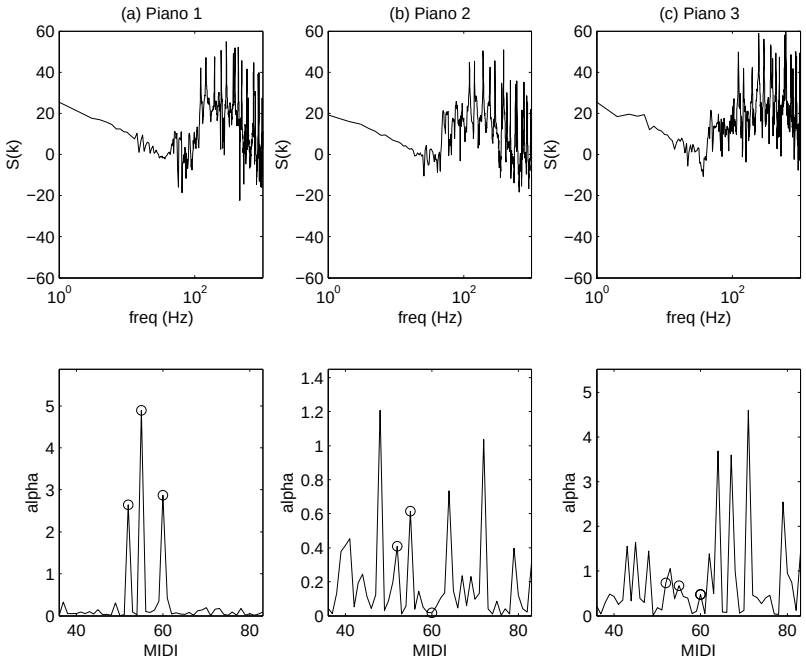

Figure 5.17: Calculation of $\boldsymbol{\alpha}$ for equal C-major chords from three different pianos. D corresponds to the first piano.

However, the availability of a database that completely matches the played sound is an ideal (and unreal) situation for real musical recordings. A useful pitch estimation method must deal with recordings for which such information is not available. Therefore, the next question is about the gen-

erality of our database (a problem equivalent, for instance, to that of the training of a neural network), or more simply: is it possible to use the same database for the analysis of any piano recordings?. The unfortunate answer is no. When dealing with chords generated by samples of the other two pianos, results were extremely poor, as shown in Fig. 5.17(b) and (c), where the system is presented with C major chords generated using *Piano 2* and *Piano 3* respectively. In fact, when studying the relative spectrum of individual notes and their mixtures, it appears that the amplitudes of partials vary significantly between different pianos (or recording conditions), thus misleading the recognition process. This can be seen at the top of Fig. 5.17.

Therefore, an adaptive approach is adopted, where the database of individual notes is first constructed from the frequency-domain analysis of the sound to be transcribed. The generated database is then used to perform the above-mentioned procedure. The proposed methodology is explained in the following sections.

5.5.4 Estimation of the database

The construction of the database from the analysed signal $s(n)$, consists of three steps: Estimation of predominant pitches in the frequency domain and selection of the fittest for the database construction, synthesis of the estimated vectors and completion of the database.

Pitch estimation in the frequency domain and selection of the best candidates

The method described in section 5.4 is used for the pre-estimation of notes present in the analysis signal. However, as the main purpose of this pre-estimation is to construct a database of individual waveforms there are some considerations to be observed.

In any practical application of a multi-pitch estimation system there are a number of parameters involved. The selection of these parameters intends to maximise the number of notes correctly identified while minimis-

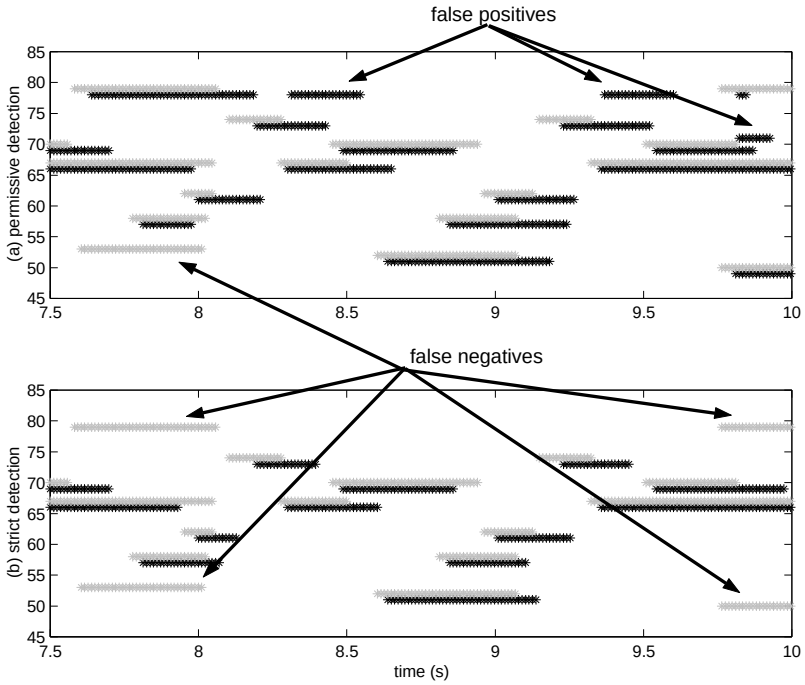

Figure 5.18: Frequency-domain detection using a permissive (a) and a strict (b) set of parameters. Results are in black and target values are in grey and slightly offset.

ing the number of false positives (estimated notes that are not present). Unfortunately, these two goals are mutually exclusive. A permissive set of parameters increases the number of identified notes, both correct and incorrect, while a strict set of parameters may have the opposite effect. An example is shown in Fig. 5.18, where a permissive and a strict detection are compared. Note that in Fig. 5.18(a) false positives appear while few present notes are not detected (false negatives). The situation is reversed in Fig. 5.18(b) where false positives are eliminated at the cost of false negatives being multiplied.

Most parameter selection is concerned with balancing these situations. However, in this particular case we are not concerned with examining all estimated notes from the frequency-domain analysis. In this case, attempt-

ing to detect all present notes in the signal is the task of the linear additive approach.

Our intention is to select, from the frequency-domain detection, only those candidates apt to construct a reliable database. The goal is to select those notes that are "undoubtedly" present in the mixture. Furthermore, considering false positives is "not-acceptable" as notes used for the construction of the database should be, in principle, perfect (unfortunately, this is not true, as shown by experimental results). For this case, the best option is the use of a strict set of parameters and the addition of further constraints to the system regarding intervals and the quantity of notes that can be selected.

Synthesis of the estimated notes

The selected notes (their pitch and times), are used to synthesise the vectors of the database. If more than one note of the same pitch is recognised, the best one is selected according to its duration, energy and rating during the estimation process. To re-synthesise the notes of the database, the implementation of the phase-vocoder analysis and synthesis explained by DeGötzen et al [GBA00] is used. The complexity of the task lies on the fact that vectors of the database correspond to individual notes while identified notes are usually not alone.

For the synthesis process only corresponding components must be kept. The peak-picking and harmonic-comb procedures implemented in section 5.4.1 are used to select the magnitudes and phases of the corresponding peaks of the estimated note within the mixture. Spectral values around those peaks, if not belonging to other relevant peaks, are also kept. This means that when notes in isolation are identified, almost all the corresponding spectrum is used for re-synthesis.

The STFT magnitudes and phases of the selected components are used to construct a modified signal in the frequency domain. Phase unwrapping is used to accurately estimate the phase of each selected bin. This recon-

structed series of spectral frames is then transformed to the time domain through the calculation of each frame's IFFT.

The resulting frames are window tapered using a Hanning window and then overlap added. Note that due to the overlap adding, the use of a Hanning window during the analysis stage introduces distortion on the synthesised signal. This distortion depends on the relationship N/R (where N is the length of the window and R is the hop size). If $N/R < 2$, the synthesised signal is amplitude modulated, causing ripple. If $N/R \geq 2$, as it is our case, the distortion is limited to a downhill effect (according to the window shape) at the boundaries of the synthesised sequence. This distortion becomes relevant for short segments. To overcome this problem, we are zero-padding the signal along the boundaries before the analysis. Finally the synthesised signal is re-scaled according to the hop size and the analysis window. The obtained signals are organised in the database D according to their pitch.

Filling the gaps

The available database at this point of the algorithm is limited to those notes detected using the frequency-domain approach. This incomplete database is not suitable for the implementation of the time-domain method. As mentioned before, the more conservative the frequency-domain estimation is (hence, the more robust its results are), the more incomplete the database, thus the more limited the capabilities of the linear additive approach.

The situation, illustrated on the left-hand side of Fig. 5.19, leaves us with a database of a few detected notes and many gaps. To deal with this, a pitch-shifting algorithm, i.e. multiplication of every frequency by a transposition factor, is used to fill those gaps. Shifting is allowed only up to half an octave from one note, to avoid large distortions.

An efficient pitch shifting algorithm can be implemented by transposing all instantaneous frequencies in each frame of the analysed note [AKZ02]. This is easily implementable within our phase-vocoder framework. A modi-

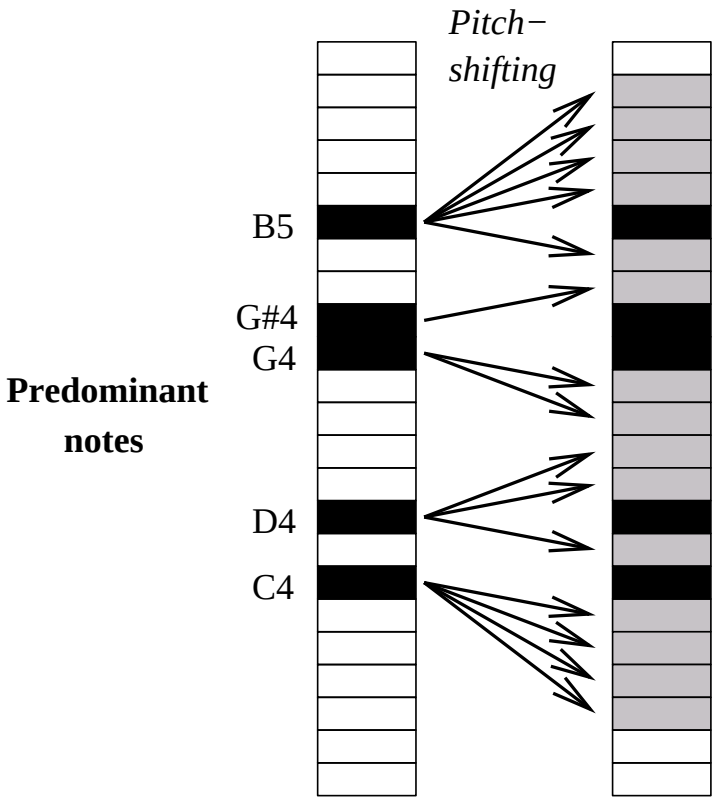

Figure 5.19: Gaps in the database are filled by pitch-shifting the predominant notes.

fied phase difference $\Delta\psi$ is estimated by multiplying the real phase difference $\Delta\varphi$ (the calculation of which is explained in section 3.2.3) by a transposition factor ρ_φ, such that:

$$\Delta\psi = \rho_\varphi \Delta\varphi \tag{5.24}$$

where:

$$\rho_\varphi = \frac{\tilde{f}_0}{f_0} \tag{5.25}$$

f_0 and $\tilde{f}_0$ are the fundamental frequencies of the original and shifted notes respectively.

The modified phase increments are used in the composing sinusoids of the reconstructed signal, thus generating the pitch-shifted versions of the original notes used to fill the gaps in D (as illustrated in the right-hand side of Fig. 5.19).

5.5.5 Integration into the blackboard framework

The algorithm for estimating pitches using the linear additive method can be sub-divided into two processes: an off-line process where the database is generated and an off-line process for the calculation of the values of $\boldsymbol{\alpha}$ and their grouping into events.

The off-line process, shown in Fig. 5.20, starts once the estimation of pitches with the frequency-domain method is done. Then, the scheduler activates the *knowledge organisation* knowledge source **KS_strict**. This knowledge source filters out the weakest estimations for each pitch, selecting the remaining detections as candidates for the individual waveforms of the database. The information about their location in the waveform, pitch and possible context is sent to the *top-down block*. **KS_select_window** moves to the selected location in the original waveform, activating the bottom-up processes (explained in chapter 3) that generate spectral magnitudes and spectral phases for each selected frame. The resulting information is used by **KS_re-synthesise** to generate an approximate waveform for the notes in question (as they are usually part of a more complex mixture).

The generated waveforms are then sent to **KS_entry_database**, a *knowledge organisation* module that compares the inputting waveform with any existing waveform of the same pitch in the **database** (a passive knowledge source). If the targeted location is not empty, this knowledge source compares both waveforms in terms of their length, energy, and average number of bins used for the re-synthesis: if a note is by itself in the original waveform, more bins are used for its reconstruction, as there is no interfering information from other notes, e.g. large peaks in unexpected positions, etc.; isolated notes are favoured over notes extracted from mixtures, hence the

higher the average of used bins, the higher the probability of survival. If the corresponding location in the database is empty, then the note is introduced without further considerations.

The whole process is automatic from the activation of **KS_strict** and only stops when there are no more candidates to be considered. At this point, and if the database is incomplete (as is usually the case), the scheduler fires **KS_fill_database**, another *knowledge organisation* module, that uses the pitch-shifting algorithm previously described to fill gaps in the database.

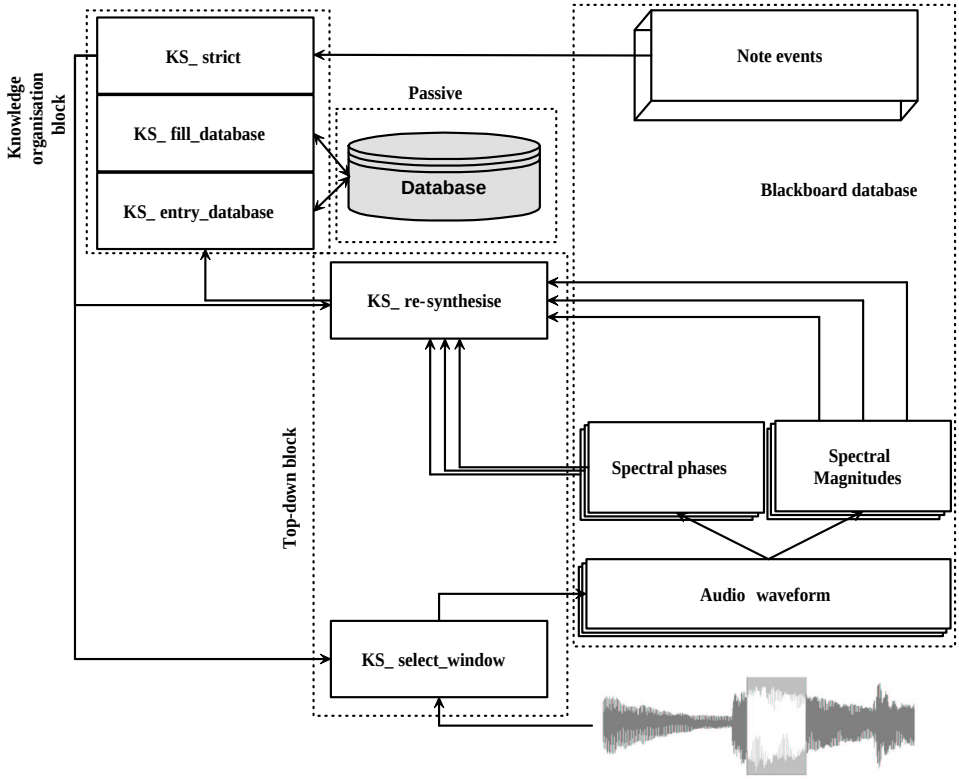

Figure 5.20: Integration of the time-domain pitch detection system into the blackboard framework: off-line process.

The completion of the database fires the on-line process of the calculation of $\boldsymbol{\alpha}$. This process is depicted in Fig. 5.21. The process occurs sequentially as with the frequency-domain pitch estimation, from beginning to end of the music file. At each time step, the information in the *audio waveform* level

of the database is used by the organisational module **KS_align** to create a database that is phase aligned with the waveform segment being analysed. The resulting **aligned database** is a passive knowledge source used solely for this instantaneous analysis. The formulae of the linear additive approach is used by the top-down **KS_alpha_calc** to generate an intermediate level in the blackboard: alpha per frame. This level is equivalent to the *frame hypotheses* level of the previous approach.

Given that notes are temporal events, the frame calculation of $\boldsymbol{\alpha}$ is not enough for pitch estimation. Again, an overlapping one second window is used by the scheduler for the grouping of α values across time. The *temporal* **KS_alpha_envelope** executes this task. It filters out spurious values from the α envelope of all considered pitches. Then it searches the remaining profile for peaks that fit the expected characteristic of a musical note: a sharp attack followed by a slow decrease. This is done by means of the analysis of the attack and decay slope of each peak.

The knowledge source checks the information for events detected using the frequency-domain approach and for onsets within the examined window. By considering the three sources of information it may decide to confirm (when an already detected note is also identified by the time-domain approach), to eliminate (when there is absence of estimation using the time-domain approach and a "weak" estimation by the frequency-domain approach) or to create (when a new note, supported by onset information, is strongly estimated by the time-domain approach). From the operation of **KS_alpha_envelope**, the definite set of results in our system is generated.

5.5.6 Results and discussion

First, two examples are shown, using piano-roll type diagrams, to illustrate the system's performance. The first of these, in Fig. 5.22 shows a segment of a Borodin piano piece. The signal was synthesised from a MIDI file (Fig. 5.22(a)) using a sampled piano sound. In general, synthesised sounds are much easier to process than real sounds. The characteristics of the

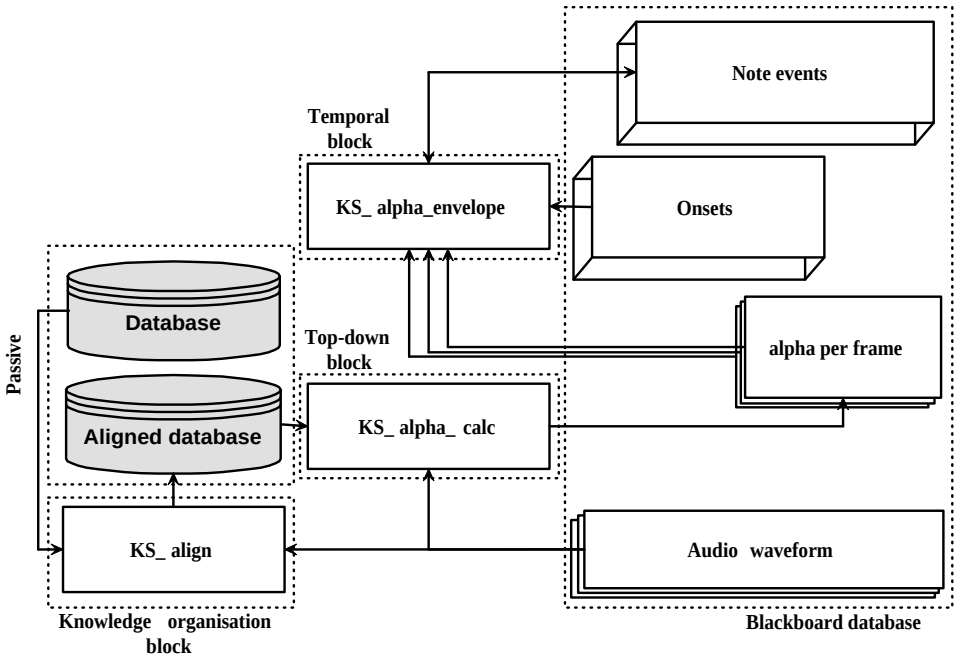

Figure 5.21: Integration of the time-domain pitch detection system into the blackboard framework: on-line process.

sound become predictable and uniform all along the musical piece. The synthesis process and our database generation process are somehow similar (generating the instrument's description from a set of basic waveforms which are manipulated to cover the entire frequency range). Thus, given that a successful recognition of notes is performed in the frequency-domain, it can be expected that the time-domain approach will perform adequately.

In Fig. 5.22(b) the frequency-domain transcription of the segment is depicted. It can be seen that all notes are identified even in complex mixtures. Some over-estimations are performed by the system, however, if the selection process is capable of filtering them out, the database will be accurately generated. In the figure "note-lines" diminish in intensity as notes decay (the higher the energy of the note, the darker in the diagram). Intensity differences between lines indicate velocity differences between notes. This energy mapping is shown here as a means to illustrate the strength of each detection. As mentioned before, the selection process chooses the strongest

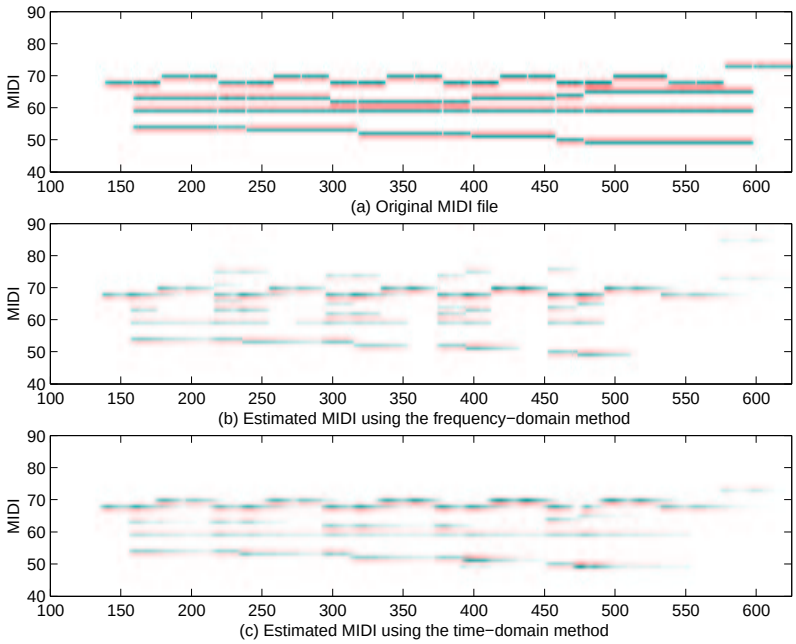

Figure 5.22: Transcription from a MIDI-synthesised piece: original (a) and estimated MIDI files using the frequency-domain (b) and the time-domain (c) approaches.

notes for the database, "darker" notes are favoured over light notes of the same pitch. Note that most over-detections are light in comparison with other notes. It seems probable that if that same note is correctly detected at some other point of the musical piece, then the time-domain system will be capable of identifying this note as an over-detection. It is important to mention that even if the figure only shows a short segment, the analysis is performed over a much larger section of the musical piece (otherwise it would be impossible for the system to accumulate enough knowledge about the instrument so as to perform well in the time-domain). In Fig. 5.22(c) the estimation using the time-domain approach can be seen. Note how most spurious values have been eliminated.

However, with real recordings, the construction of the database is less

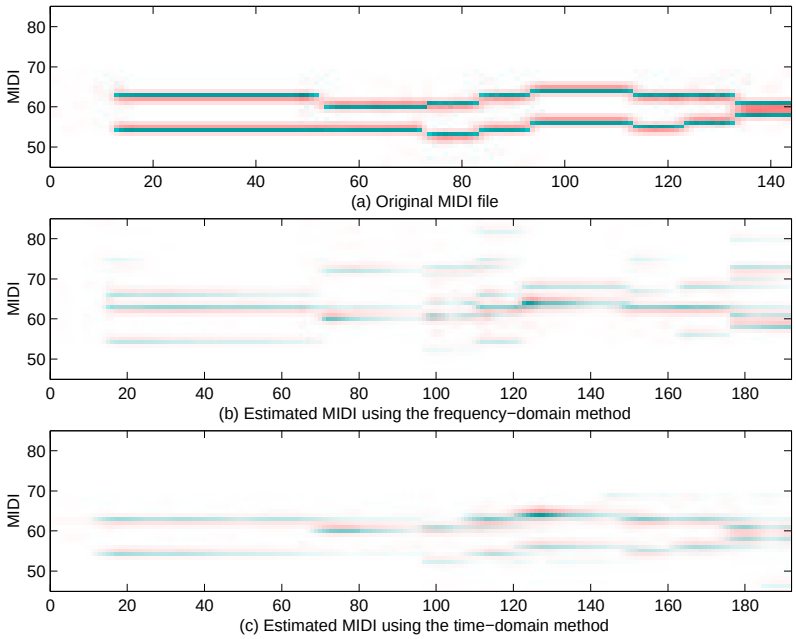

Figure 5.23: Transcription from a real recording: original (a) and estimated MIDI files using the frequency-domain (b) and the time-domain (c) approaches.

effective. Fig. 5.23 depicts the performance of both systems when dealing with real recordings. The difficulty of the process is made evident by looking at the proliferation of errors in both Fig. 5.23(b) and Fig. 5.23(c). This is true even with low polyphonies, such as the one depicted (although intervals in the segment are fifths, which as we mentioned several times before, are not easy to identify).

With real recordings the database is more likely to be corrupted, generating miss-detections with the linear-additive approach. Furthermore, even when notes are correctly selected for the database, they do not necessarily fit nicely all the executions of the same note during the length of the performance. Waveforms of the same note are subject to significant variations depending on the energy at which they have been played, their length,

Composer	% DET	% FN	% OK	% FP
Mozart	82.20	17.80	80.56	19.44
Beethoven	75.16	24.84	78.51	21.49
Debussy	82.32	17.68	79.78	20.22
Joplin	64.59	35.41	70.93	29.07
Ravel	72.29	27.71	66.53	33.47
TOTAL	76.96	23.04	77.67	22.33

Table 5.5: Note estimation results using the linear additive approach

the work of the pedals, the context (the interaction with other notes being played at the same time), all of which is neglected by the linear additive approach. Notes generated by synthesised sounds are effectively independent of each other, something that cannot be said about notes generated from the complex mechanism of a grand piano. Yet, as shown by Fig. 5.23(c), the time-domain transcription is usually not far from the original. Let us quantify how far.

Table 5.5 shows rates for the estimation of pitches on the test-bed introduced in section 5.4.4, using the linear additive approach on the time-domain. By comparing with Table 5.1, it can be seen that the overall rate of good detections has increased by 8%, up to almost 77%. This is a sharp increase quantitatively speaking, and even more so if considering that the higher the rate of good detections the more difficult is to improve it (as only very difficult detections, e.g. notes in complex polyphonies, remain to be identified). In Table 5.6, the decrease in false negatives is categorised by the most common sources of errors. It can be seen how for every file the amount of false negatives is consistently decreased for harmonic errors. The contribution of *repeated notes* remains similar (very low).

The increase in good detections is notably sharp (about 10%) for Mozart's and Beethoven's music, and much less so for Joplin's and Ravel's music (nearly 4%). This is puzzling, if considering that, the latter having

MOZART		BEETHOVEN	
False negatives = 17.80 %		False negatives = 24.84 %	
Octave (higher)	1.73 %	Octave (higher)	3.88 %
Octave (lower)	0.45 %	Octave (lower)	0.54 %
Thirds	2.87 %	Thirds	3.18 %
Fifths	2.94 %	Fifths	1.63 %
Repeated notes	0.38 %	Repeated notes	3.49 %

DEBUSSY		JOPLIN	
False negatives = 17.68 %		False negatives = 35.41 %	
Octave (higher)	2.42 %	Octave (higher)	9.57 %
Octave (lower)	0.76 %	Octave (lower)	7.81 %
Thirds	2.04 %	Thirds	0.32 %
Fifths	1.91 %	Fifths	2.87 %
Repeated notes	0.38 %	Repeated notes	0.00 %

RAVEL		**TOTAL**	
False negatives = 27.71 %		False negatives = 23.04 %	
Octave (higher)	2.60 %	Octave (higher)	3.71 %
Octave (lower)	2.16 %	Octave (lower)	1.71%
Thirds	0.43 %	Thirds	2.30 %
Fifths	3.46 %	Fifths	2.37 %
Repeated notes	0.00 %	Repeated notes	1.24 %

Table 5.6: Time-domain method: categorisation of false negatives per composer

a lower good detections' rate in the frequency-domain, the opportunity for increasing the correct estimations was higher than with the other files.

The answer lies on the recall rates for both methods. In Table 5.1, the lowest recall rates were for Joplin's and Ravel's music. The high number of false positives increases the chance for corruptions in the construction of the database. The selection process is designed to attempt to overcome this, but this is not always possible. The higher the false positives rate, the higher the chance of errors in the database.

In general, the reliability of all estimations is affected by the use of the time-domain approach as can be seen from the recall and false positives columns of Table 5.5. The overall increase in the false positives' rate is of 6%. It is particularly critical for Joplin's music, and more or less uniform for all other files. This is a price that we need to pay for the increase in good estimations (hypothesises and expectations are always more unreliable than bottom-up observations). However, in the case of Joplin's and Ravel's music, the gain and loss relationship is not favourable. All this seems to support the idea that there is a minimum level of "correctness" of detections using the frequency-domain approach that we need to achieve in order to make the best out of the linear additive approach. Unfortunately we do not have this information prior to analysis, further complicating the decision-making process. It is also important to mention that both Joplin's and Ravel's test-beds are shorter than that of the other authors. As quantities of information are essential for the correct construction of high-level knowledge, this may be a factor contributing to the increase or decrease in the accuracy of the detection.

Table 5.7 categorises false positives according to the most common types. Harmonic errors are still strong in the new detections, basically due to corrupt information in the database, but also because of the organisation of harmonic information in the solution space: If we think about individual notes as independent vectors in the solution space, we will find that harmonically related vectors are closer. As the analysed sound moves into the

MOZART		BEETHOVEN	
False positives = 19.44 %		False positives = 21.49 %	
1^{st} harmonics	4.14 %	1^{st} harmonics	5.03 %
Sub-harmonics	1.70 %	Sub-harmonics	3.65 %
Thirds	3.03 %	Thirds	2.35 %
Fifths	2.14 %	Fifths	3.08 %
Repeated notes	1.18 %	Repeated notes	1.05 %

DEBUSSY		JOPLIN	
False positives = 20.22 %		False positives = 29.07 %	
1^{st} harmonics	3.33 %	1^{st} harmonics	2.98 %
Sub-harmonics	4.56 %	Sub-harmonics	6.13 %
Thirds	3.45 %	Thirds	2.45 %
Fifths	3.33 %	Fifths	5.78 %
Repeated notes	0.62 %	Repeated notes	0.70 %

RAVEL		**TOTAL**	
False positives = 33.47 %		False positives = 22.33 %	
1^{st} harmonics	6.37 %	1^{st} harmonics	4.22 %
Sub-harmonics	6.77 %	Sub-harmonics	3.72 %
Thirds	1.20 %	Thirds	2.73 %
Fifths	8.37 %	Fifths	3.51 %
Repeated notes	1.59 %	Repeated notes	0.10 %

Table 5.7: Time-domain method: categorisation of false positives per composer

solution space, we measure its distance to the database vectors. Depending on phase and energy non-linearities (not considered by this approach), this movement can occur in the mid-space between two neighbouring vectors, causing the alternate activation of the corresponding $\boldsymbol{\alpha}$. From experimentation, this kind of alternation does not seem to be the rule. However, as vectors get too close, or as the particular "spatial movement" of a given note becomes too large (when notes are long for instance), the probability of having problems of this sort increases.

Although, we have concentrated on describing the practical issues that arise when using the time-domain approach, it is undeniable that there is improvement on the performance of the system (the trade-off between good detections and recall in Table 5.5 is better than in Table 5.1). Mozart's and Beethoven's experiments where the longest in duration, and also the ones with the sharpest improvement of the whole test-bed. This supports the case that more experience (and hence more reliable knowledge) only improves the accuracy of the detections.

Interestingly, the biggest limitation of the time-domain approach seems to be its dependency on the frequency-domain approach in order to acquire the necessary knowledge for detection. As the theory showed, and the known-database results exhibited, the capabilities of the system are very high given the reliability of the database. This is confirmed by the fact that the worst results were obtained with those files that showed the highest number of false positives during the frequency-domain estimation process.

Although the importance of the contribution of top-down processing is doubtless, bottom-up processing is still the backbone of perception (or simulated perception). Without reliable sensorial information, it is not possible to build useful experiencing knowledge. These experiments are helpful to corroborate this.

5.6 Summary

Initially we proposed that pitch is the main component of a musical note, and hence in our limited transcription procedure the key element that needs to be estimated. We defined pitch to be the tonal frequency that provides the better fit between a recorded tone and the note in an hypothetic score that originated it in the first place. A distinction was made between the case in which these pitches come individually (monophonic) and when they are part of a bigger context, in which they exist simultaneously (polyphonic). We concentrate in this chapter on the calculation of pitch in polyphonic music.

Several methods are proposed in the literature based on clustering of frequency-domain information, the use of external knowledge, the search for an optimal representation and the use of statistics to describe music. We use these previous experiences to better understand the complexities of the polyphonic pitch estimation process (emphasising the problems of harmonicity and polyphony), and to create the first system based on the analysis of frequency-domain information.

The proposed method analyses the signal on a frame-by-frame basis, detecting the most prominent spectral peaks and designing a set of target harmonic combs based on the information provided by those peaks. The feasibility of these harmonic combs, given the current spectrum, is analysed and evaluated using a set of individual and competitive rules. Success in complying with these rules creates a set of note hypotheses associated with the current frame.

By extending the analysis to a long window of several frames, a set of hypothetic data is generated that acts as a first approximation to explaining the semantics of the music being played during that segment. The system analyses consistency and reliability of produced hypotheses along the time axis. Another set of rules is applied individually to each pitch strip within the window to evaluate the possibility of its existence. Events that are confirmed using this analysis are then used as the output of the frequency-

domain system.

The integration of this method into the blackboard framework is explained and results on real recordings are shown and discussed. It is illustrated how nearly 70% of notes can be identified by the system with high reliability. Mistakes are often associated with the common constraints in frequency-domain approaches: those related to high polyphonies, short durations and harmonic intervals. An alternative evaluation is mentioned, were estimations produced by the system are used with success to retrieve scores from a large database (see Appendix A).

The findings of the first approach lead us to conclude that more knowledge is needed to perform the estimation task with greater success. The recreation of the acquisition of this knowledge through experience, and its use for a high-level task such as this, generated the approach explained in the second part of this chapter.

A new method is presented that identifies notes from polyphonic mixtures in the time-domain. It helps to overcome common issues that arise when using frequency-domain transcription for polyphonic music. The approach assumes short segments of the original waveform to be the linear sum of weighted individual waveforms (corresponding to the individual notes of the played instrument). The theory is developed to lead to the conclusion that, by estimating the values of the mixing vector (denominated $\boldsymbol{\alpha}$) for each frame, accurate polyphonic pitch detection can be achieved. This is true provided that we have the original waveform and that we have a database of waveforms corresponding to individual notes of the instrument.

However, two conditions need to be satisfied, phase-alignment, obtained through independent shifting of each vector, and a reliable database, constructed by using the results of the frequency-domain approach. The self-generating database stands for expert knowledge, boosting the robustness of the method even with complicated polyphonies or recording conditions.

The integration of this method into the blackboard is explained and results on real recordings are shown and discussed. The method improves

the good detections' rate by 8%, but with a cost to the reliability of the detection of nearly 6% of the estimations. Detailed analysis concludes that a considerable improvement to the previous approach can be obtained when the frequency-domain method achieves a minimum level of accuracy. Otherwise, incorrect detections leak into the construction of the database badly affecting the performance of the second system. This is consistent with the dependence of high-level processing on the reliability of sensorial information and basic processing.

Chapter 6

Conclusions and Perspectives

6.1 Conclusions

A system has been introduced for the task of automatic analysis of simple polyphonic music. It uses the blackboard architecture as a realisation of the interactive approach to perception, allowing the combination of bottom-up and top-down processing.

The system estimates the individual features of musical notes from audio waveforms, finally combining them into note events used to render a MIDI representation of the music being played.

The phase-vocoder was implemented to produce a time-frequency representation of the signal. It is able to return an accurate estimation of the features (amplitude, phase and frequency) of the sinusoids involved in the Fourier analysis of the signal. This information is the basic data on which all estimations (and observations leading to those estimations) are based.

It was preferred over other representations for a number of reasons: For the polyphonic pitch estimation task, high frequency resolution is needed at the higher end of the spectrum as much as at its lowest end (as resolution between partials is as important as resolution between fundamentals). Frequency resolution is preferred to time resolution. For this reason we favoured an evenly-spaced filter-bank over a constant-Q representation. The phase-

vocoder can be easily implemented using the fast Fourier transform (FFT) algorithm, minimising the computational expense of our signal analysis. As it is a well-known technique, developments can be made using its theoretical basis (this is in fact the case with our onset detection algorithm).

The importance given to phase information, within the context of the phase-vocoder, is used to our advantage during feature extraction. Based on an algorithm for transient and steady-state separation, an original method is proposed for detecting onsets in a music file.

The main suggestion is that the phase change between two consecutive frames of our sinusoidal analysis should remain constant if the corresponding sinusoid is in its steady-state. This is equivalent to calculating a simple measure of angular differences between three consecutive frames, which remains close to zero during the steady-state and increases otherwise. The distribution of those bin-by-bin angular differences was analysed, concluding that attack transients are related to spread distributions, while the beginning of the steady-state is related to peaky (lepto-kurtic) distributions. These observations are quantified by calculating the inter-quartile range and the kurtosis coefficient of consecutive distributions. The resulting functions are used for onset detection.

Peaks are detected in these functions using a weighted-median approach over a long window. The optimal weight was experimentally selected to maximise results. The experiments, made on a test-bed of real CD recordings including complex pop and jazz files, show very high detection rates with a low number of false positives. These false positives increase with the complexity of the signal, usually due to distortions in the phase information. The percussiveness and sibilance produced by the sung voice also affect results negatively. However, results show improvement over traditional energy-based onset detection systems that are not able to perform successfully in such complex mixtures.

The main task faced by the system is the estimation of pitches within polyphonic mixtures. A method is proposed to perform this task by analy-

sing information in the frequency-domain. First, the method searches for high peaks on each frame spectrum. Selected peaks are assumed to be one of the first few harmonics of a hypothetic note. Each assumption is used to build a target harmonic comb which has lobes situated in the expected positions of the corresponding notes. These combs are evaluated with regard to their relationship with the actual spectrum using individual and competitive rules. Winning hypotheses are selected for each frame and then grouped in time to construct note events. Successful events need to be reliable and consistent during the length of a minimum duration window.

Experiments were made on real recordings of polyphonic MIDI files of piano music. By comparing original and resulting MIDI files, the performance of the system was quantified. It is shown that the quality of the estimations is affected by the polyphony of the files: the higher the number of notes, the lower the chance of successful estimations; and by the contents of those polyphonies: i.e. harmonically related intervals. The short duration of notes also contributes to the proliferation of errors.

Although imperfect, results were used for the task of music information retrieval with success (see Appendix A). This shows that even limited descriptions of the signal's musical contents can be put to good use in certain applications.

The method's performance is constrained by the limitations of the common approach of analysing information in the frequency-domain (where certain intervals cannot be properly differentiated), as well as the lack of high-level knowledge within the system. An alternative approach is introduced that tackles those disadvantages by using knowledge about the played instrument in the form of a note's database, and by performing its analysis in the time-domain.

The method assumes a signal's segment to be the weighted sum of the waveforms of the individual notes of the played instrument. The estimation process is reduced to the calculation of the weighting vector given the sound and the database. The implementation requires dynamic phase-alignment

between sound and database for each analysis frame.

It is demonstrated how the database needs to be an accurate representation of the played instrument for the method to be successful. The use of a generic representation of the instrument does not return valuable results. As the waveforms of individual notes is not usually available for real recordings, it was proposed that the database could be constructed from the estimations of the frequency-domain approach.

Results based on real polyphonic piano music showed improvement upon the initial frequency-domain approach. It is demonstrated that the improvement is significant when provided with a reliable frequency-domain estimation. However, results are less encouraging when the uncertainty of frequency-domain estimations is high. The dependency of the time-domain approach on the initial frequency-domain estimations is its greatest weakness. As with human perception, obtaining knowledge and experience depends upon the bottom-up processing of sensorial information. If this processing is corrupted in the first place, the knowledge obtained is of no use for future estimations. On the other hand, if the knowledge is reliable, then perception becomes robust even in complicated environments (e.g. high polyphonies of harmonically related intervals in real recordings).

Finally, the results obtained illustrate the capabilities of the blackboard architecture as a framework that supports the combination of different types of processing, knowledge and information. The "updating" of the framework, at the end of each sub-system's explanation, shows the flexibility and expandability of the current architecture.

6.2 Perspectives

An immediate suggestion for further development is related to the linear additive approach and the process of estimating the database. A statistical approach could be used to generate the database, for example searching for independent components that correspond to note waveforms. Another option would be to search the signal, at the positions of notes estimated

using the frequency-domain approach, for common components, and to use those components to represent the target note.

An interesting direction would be to filter the differences between different pianos (or different recordings), stripping the waveforms to their basic, common features. If successful, this would allow the design of a general database for a single instrument, altogether eliminating the uncertainty produced by the database estimation.

In general, future research should concentrate on the next level of the music structure, that of tempo, metre and harmony. Note onset and pitch information can be used to that end. Successful mapping of the available data into these levels would produce a representation closer to scores (approaching the process of a real transcription task). Moreover, the retroactive effect that such high-level estimations can have on the detection of notes is an interesting field of study.

Experimental results suggested that the use of knowledge increases the chances of good detection. Therefore, the development of algorithms that expand this knowledge-base to higher levels of complexity is the natural extension of the research presented in this dissertation.

Appendix A

Paper Reprint

This article was originally published as:

Pickens, J., Bello, J.P., Crawford, T., Dovey, M., Monti, G., Sandler, M. and Byrd, D. *Polyphonic Score Retrieval Using Polyphonic Audio Queries: A Harmonic Modelling Approach.* In Proceedings of the 3rd International Conference on Music Information Retrieval (ISMIR)[1], Paris, France. pages 140-149. October, 2002.

References are included in the main bibliography.

[1]Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page.

Polyphonic Score Retrieval Using Polyphonic Audio Queries: A Harmonic Modeling Approach

Jeremy Pickens[†], Juan Pablo Bello[‡], Tim Crawford[♯],Matthew Dovey[♭], Giuliano Monti[‡], Mark Sandler[‡] and Don Byrd[♮]

[†]University of Massachusetts, Amherst

[‡]Queen Mary, University of London

[♯]King's College, London

[♭]Oxford University

[♮]Indiana University, Bloomington

Abstract: This paper extends the familiar "query by humming" music retrieval framework into the polyphonic realm. As humming in multiple voices is quite difficult, the task is more accurately described as "query by audio example", onto a collection of scores. To our knowledge, we are the first to use polyphonic audio queries to retrieve from polyphonic symbolic collections. Furthermore, as our results will show, we will not only use an audio query to retrieve a known-item symbolic piece, but we will use it to retrieve an entire set of real-world composed variations on that piece, also in the symbolic format. The harmonic modeling approach which forms the basis of this work is a new and valuable technique which has both wide applicability and future potential.[2]

1 Introduction

Music information retrieval is a rapidly growing field. As more music collections come online, the demand to search these collections increases. Music collections, or sources, exist in one of two basic formats: audio and symbolic. To complicate matters, music queries exist in both formats as well.

[2]This work was supported in part by the Center for Intelligent Information Retrieval and in part by NSF grant #IIS-9905842. Any opinions, findings and conclusions or recommendations expressed in this material are the author(s) and do not necessarily reflect those of the sponsor.

A comprehensive music retrieval system should be able to allow queries in either format to retrieve music pieces in either format. The problem lies in the fact that the features readily available from audio files (MFCCs, energy) do not correspond well with the features available from symbolic files (note pitches, note durations) It is a "vocabulary mismatch" problem.

Our system will bridge the gap between audio and symbolic music using transcription algorithms together with harmonic modeling techniques. In this manner we allow users to present queries in the audio format and retrieve pieces of music which exist in the symbolic format. This is one of the earliest goals of music retrieval, and until now it has only been possible within the monophonic domain. We extend the realm of possibility into the remarkably more difficult polyphonic domain, and show this through successful retrieval experiments for both known-item and variation queries. The ability to use polyphonic audio queries to retrieve pieces of music from a polyphonic symbolic collection is a major step forward in the field.

The remainder of this paper proceeds as follows: In Section 2 we give a brief review of the problem domain and existing literature. Section 3 locates this paper within the larger framework of the "language" modeling approach to Information Retrieval. Section 4 contains an overview of our system. In Section 5 we explain our audio music transcription techniques. In Section 6 we explain our harmonic modeling techniques, while in section 7 we show how two models are compared for dissimilarity. Finally, Sections 8 and 9 contain our experimental design, results, discussion and conclusion.

2 Background and Related Work

To date, research in the field of ad hoc music retrieval has experienced two fundamental divisions. The first division is one of representation. Music may either be presented as a performance or as instructions to the performer. A performance is an audio file, in a format such as WAV or MP3. Instructions to the performer exist in a symbolic format, either as a MIDI file

(www.midi.org) or in Conventional Music Notation (CMN) format [AMN01], both of which express some manner of instructions about what notes should be played, when, for how long, and with what instrument or dynamic.

This division between actualized performance and instructions for a performance manifests itself in the types of features readily extractable from digital forms of audio and symbolic music. Those retrieving audio tend to work with features such as MFCCs, LPCs, centroids, or energy, while those retrieving symbolic sources use actual note pitch and/or duration, as these values are known.

The second division in music IR is one of complexity, or monophony versus polyphony. Monophonic music has at most one note playing at any given time; before a new note starts the previous note must have ended. Polyphonic music has no such restrictions. Any note or set of notes may begin before any previous note or set of notes has ended, which proves difficult for any clear, unambiguous sense of sequentiality. Therefore, techniques which work for monophonic music, such as string matching or n-gramming, are more difficult to apply to the polyphonic domain. Furthermore, reasonably accurate conversions from audio to symbolic music is generally seen as a solved (or at least manageable) problem for monophonic music, but still a fairly inaccurate, unsolved problem for polyphonic music.

Polyphonic music in general is more complex and difficult to work with. Indeed, some of the earliest works in music retrieval remained entirely within the monophonic domain [GLCS95, MSBW97]. These "query by humming" systems allow the query to be presented in audio format, and then converted to symbolic format to be used for query on a monophonic symbolic collection. Gradually, systems which allowed monophonic queries upon a polyphonic collection, a more difficult prospect, were introduced [BPMS02, LT00, UZ99]. The query is still monophonic, so conversion of the query between audio and symbolic formats remains possible. The collection to be searched may therefore be audio or symbolic, as the query may easily be converted in either direction to match. But again, this is only possible

because the query is monophonic.

Most recently, polyphonic queries upon a polyphonic collection have become possible. Yet because of the complex nature of polyphonic music and the difficulty of accurate conversion, researchers tend not to mix the audio and symbolic domains. Research has either focused on polyphonic audio queries upon polyphonic audio collections [Foo00, PBO00, TEC01], or polyphonic symbolic queries upon polyphonic symbolic collections [BD85, Dov99, DR01, MWL01, PC02]. We know of no prior work which tackles polyphony, audio, and symbolic music all in the same breath.

Of the papers mentioned above, the one that most closely resembles our work is Purwins et al [PBO00]. These authors have devised a method of estimating the similarity between two polyphonic audio music pieces by fitting the audio signals to a vector of key signatures using real-valued scores, averaging the score for each key fit across the entire piece, and then comparing the averages between two documents. As do we, these authors use Krumhansl distance metrics [Kru90] to assist in the scoring. One of the main differences, however, is that these authors attempt to fit an audio source to a 12-element vector of keys, while we fit a symbolic source to a 24-element vector of major and minor triads. Furthermore, by averaging their key-fit vector across the entire piece, their representation is analogous to our 0^{th}-order Markov models. Our paper utilizes not only 0^{th}-order models, but 1^{st} and 2^{nd}-order models as well. Moreover, the Purwins paper was not specifically developed as a music retrieval task, and thus has no retrieval-related evaluation. We present comprehensive known-item as well as recall-precision results.

Finally, a paper by Shmulevich et al [SYHC$^+$01] also uses some of the same techniques presented here, such as Krumhansl's distance metrics and the notion of smoothing, the latter which will be presented in section 6.2. The domain to which these techniques are applied are monophonic, but Shmulevich's work nevertheless demonstrates that harmonic analysis and probabilistic smoothing can be valuable components of a music retrieval

system.

3 Language Modeling Approach

Language Modeling (LM) has received much attention recently in the text information retrieval community. It is only natural that we wish to leverage some of the advantages of LM and apply it to music. Ponte explains some of the motivations for this framework:

> [A language model is] a probability distribution over strings in a finite alphabet (page 9)... The approach to retrieval taken here is to infer a language model for each document and to estimate the probability of generating a query according to each model. The documents are then ranked according to these probabilities (page 14)...The advantage of using language models is that observable information, i.e., the collection statistics, can be used in a principled way to estimate these models and do not have to be used in a heuristic fashion to estimate the probability of a process that nobody fully understands (page 10)...When the task is stated this way, the view of retrieval is that a model can capture the statistical regularities of text without inferring anything about the semantic content (page 15)." [Pon98]

Even though our retrieval task is polyphonic music rather than text, we are duplicating the LM framework by creating statistical models of each piece of music in a collection and then ranking the pieces by those statistical properties. Thus, while it might be more appropriate to name this work "statistical music modeling", we still say that we are taking the language modeling *approach* to information retrieval. So rather than attempting a formal analysis of the harmonic structure of music, we instead "capture the statistical regularities of [music] without inferring anything about the semantic content".

Nothing illustrates this more than our choice, explained in section 6, to characterize the harmony of a piece of music at a certain point as a *probability distribution* over chords, rather than as a single chord. Selecting a single chord is akin to inferring the semantic meaning of the piece of music at that point in time. While useful for some applications, we feel that for retrieval, this semantic information is not necessary, perhaps even harmful if the incorrect chord is chosen. Rather, we let the statistical patterns of the music speak for themselves.

To our knowledge, the first LM approach to music IR was done in the monophonic domain [Pic00]. Other recent techniques, which also take the LM approach (though without always explicitly stating it), apply 1^{st}-order Markov modeling to monophonic note sequences [RB01, HRG01]. Further work extends the modeling to the polyphonic domain, using both 0^{th} and 1^{st}-order Markov models of raw note simultaneities to represent scores[BDW$^+$01].

4 System Overview

The goal of this system is to take polyphonic audio queries and return polyphonic symbolic pieces of music, highly ranked, which are relevant to the given query. This is done in a number of stages, as outlined in Figure A.1.

Offline and prior to query time, the entire source collection (the set of polyphonic scores which are to be searched) is passed through the harmonic modeling module, described in Section 6. Each piece of music, each document, is then "indexed", or stored, as a model. At query time, the system is presented with polyphonic audio, such as a digitized recording of a piano piece from an old LP. The query is first passed through the audio transcription module, described in Section 5. The transcription from this module is passed to the harmonic modeling module, and a model for the query is created.

Finally, a scoring function is used to compare the query model with each of the document models, and give each query-document pair a dissimilarity

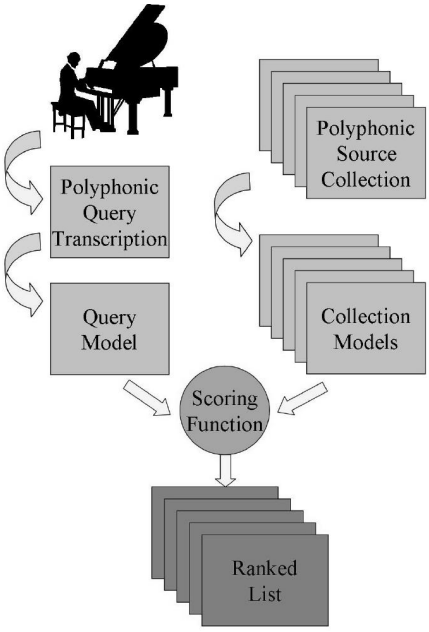

Figure A.1: System Overview

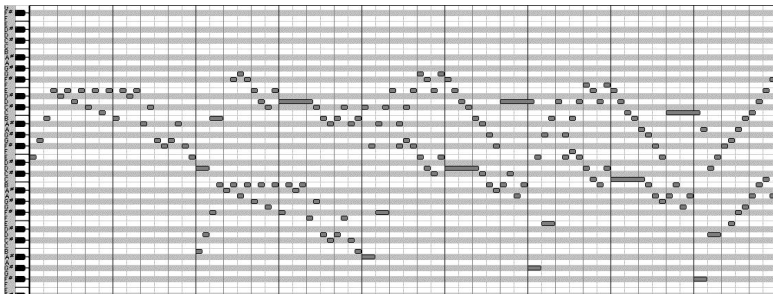

Figure A.2: Bach Fugue #10 Original Score

value. Documents are then sorted, or ranked, by that value, with the least dissimilar at the top of the list.

5 Audio Transcription

Automatic music transcription is the process of transforming a recorded audio signal into a representation of its musical features. We will limit our definition to the estimation of onset times, durations and pitches of the

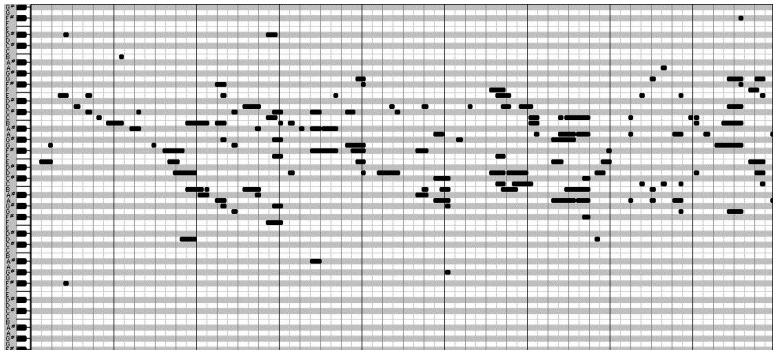

Figure A.3: Bach Fugue #10 from Polyphonic Transcription II algorithm

notes being played. This task becomes increasingly complicated when dealing with polyphonic music because of the multiplicity of pitches, inconsistent durations, and varied timbres. Most monophonic transcription techniques are therefore not applicable. In fact, despite several methods being proposed with varying degrees of success [Dix00a, Mar00a, Kla98a, Mar96a], automatic transcription of polyphonic music remains an unsolved problem.

We offer two figures as an example of this transcription procedure. Figure A.2 is the original score of Bach's Fugue #10 from Book I of the Well-tempered Clavier, presented here in piano-roll notation. A human musician then performs this piece, and the audio signal is digitized. Figure A.3 is the transcription of this digitized audio from one of our algorithms. It is with this imperfect transcription that we still achieve excellent retrieval results.

We locate the audio transcription task within the context of Computational Auditory Scene Analysis (CASA). In this context, systems try to explain the analysed signal following a set of perceptual rules and sound models. These rules suggest how to group the elements from the signal time-frequency representation into auditory objects (i.e. musical notes). In polyphonic music, events overlap both in the time and the frequency domain, meaning that transcription systems should be able to analyse the signal in both domains in order to return an accurate representation of the scene. From this approach we propose two different methods. Both techniques will be used, separately, to produce queries, and retrieval results for each tran-

scription technique will be given. We do this to show that our harmonic modeling algorithm is robust to varying transcriptions and their associated errors.

5.1 Polyphonic Transcription I

Our first method is an extension and reworking of a technique used for monophonic transcription in Monti [MS02]. Fourier analysis is used to represent the signal in the frequency domain. An auditory masking threshold is calculated using a perceptual model. Only spectral maxima above such a threshold are chosen to represent the signal. The Phase-Vocoder technique is used to calculate the instantaneous frequencies of the peaks, by interpolating the phase of two consecutive frames. The analysis is optimised for the steady state part of the notes.

Once the representation of the signal is given as a set of spectral peaks, the system groups the peaks according to their frequency position and time evolution. The grouping rules are: harmonic relation in the frequency domain and common onset in the time domain. For the implementation of these rules, which group peaks into objects (notes) we used the Blackboard model [EM88]. This model has shown great flexibility and modularity, which is important when implementing additional rules.

The system starts selecting the lowest available frequency peak and, assuming it to be a note's fundamental, looks for harmonic support among the other peaks. The support of a note hypothesis is given by a fuzzy rate depending on the fundamental frequency position and energy, and the harmonic support in the spectrum. If the note is confirmed as an hypothesis, its harmonic peaks are eliminated from the hypothesis space so they cannot be chosen as new fundamental hypotheses. However, they still may contribute to other notes' hypotheses since the partials of the notes composing a chord often overlap in western music.

The algorithm iterates while there are peaks in the spectrum. Hypotheses qualify as note objects, only if they last in time for a minimum number

of (activation) frames. Once a note is recognised the system predicts its evolution in the spectrum, and in future analysis the existing notes are verified before searching for new notes. If the spectrum reveals change in the frequencies' positions or amplitude the system formulates new note hypotheses corresponding to the new events detected. Using this method, octave errors are eliminated, but at the cost of failing to detect octave intervals when played simultaneously. The system extracts onsets, offsets and MIDI pitches from the audio and writes them in a MIDI file for listening and retrieval tasks.

5.2 Polyphonic Transcription II

Our second system is an extension of work found in Bello [BDS02, BS00]. We again begin by apply Fourier analysis on overlapping frames in the time-domain. The phase-vocoder technique is also used to estimate the exact instantaneous frequency value for each bin in the frequency-domain representation. However in this approach all frequency peaks are used, regardless of their perceptual conditions.

Two levels of hypotheses are considered here. On each analysis frame, all musical notes within the evaluated range (from 65 to 2kHz) are considered to be 'frame' hypotheses. Associated with each of these frame hypotheses a filter is developed in the frequency domain. To do this we assume that a note with fundamental frequency f_k must (theoretically) present frequency partials located according to:

$$f_{m,k} = m \cdot f_k \sqrt{1 + (m^2 - 1) \cdot \beta_k} \tag{A.1}$$

where β_k is the inharmonicity factor (note and instrument dependent) [FR91], and $m = 1 \ldots M$, with M such that $f_{M,k} \leq f_s/2$. The filter associated with f_k behaves like a comb filter with lobes centered at the expected partials' frequencies and bandwidths equal to half the tone-distance between the hypothetical note and its closest neighbour (a quarter or half a tone depending on the note).

The frame's frequency-domain is processed through this filter-bank, producing a group of spectrums associated with each of the frame-hypotheses. The hypotheses are rated according to the ratio between the filtered spectra energy and the energy of the original spectrogram. Hypotheses with high ratings are classified as 'note' hypotheses and followed over time. If continuity and envelope conditions are satisfied, then the note is recognised as a note-object of the signal.

Note that in this approach, no onset detection is performed on the audio signal. Timing information depends on the behaviour of the instantaneous rating of each possible note. A smoothing window is used to group events that are very close in time.

An important difference from the previous approach is that frame hypotheses are evaluated independently, allowing any interval to be detected. This brings as a consequence the detection of octave intervals and the proliferation of octave-related errors. As with the previous transcription algorithm, the system extracts onsets, offsets and MIDI pitches from the audio and writes them in a MIDI file for listening and retrieval tasks.

6 Harmonic Modeling

A harmonic model is our term for a Markov Model in which the states of the model are musically salient, harmonic entities. The process of transforming polyphonic music into a harmonic model divides into three stages. In the first stage, *harmonic description*, the music document to be modeled is broken up into sequences of note sets, and each of those note sets are fit to a probability vector. Each of these note sets is assumed to be independent of the neighboring sets. This assumption, while necessary for the modeling, is not always accurate, in particular because harmonies in a piece of music are often defined by their context. The second stage of the harmonic modeling process is therefore a *smoothing* procedure, designed to account for this context. Finally, the third stage is the process by which *Markov models* are created from the smoothed harmonic descriptions. Stages one and three are

covered in greater detail in [PC02], while stage two is a new technique first described in this paper.

6.1 Harmonic Description

Recall from Section 1 that polyphonic music has no innate, one-dimensional sequence. Arbitrary notes or sets of notes may start before the current note or set of notes has finished playing. It therefore becomes necessary for us to artificially impose sequentiality. This is accomplished by ignoring the played duration for every note in a score, and then selecting at each new note onset all the notes which also begin at that onset. These event-based sets are then reduced, mod 12, to octave-equivalent pitch classes and given the name *simultaneity*.

We define a *lexical chord* as a codified pitch template. Of the 12 octave-equivalent (mod 12) pitches in the Western canon, we select some n-sized subset of those, call the subset a *chord*, give that chord a name, and add it to the lexicon. Not all possible chords belong in a lexicon; with $\binom{12}{n}$ possible lexical chords of size n, and 12 different choices for n, we must restrict ourselves to a musically-sensible subset. The chord lexicon will furthermore make up the state space of our Markov model, in addition to providing the basis for the harmonic description.

The chord lexicon used in this paper is the set of 24 major and minor triads, one each for all 12 members of the chromatic scale: C Major, c minor, C♯ Major, c♯ minor ... B♭ Major, b♭ minor, B Major, b minor. No distinction is made between enharmonic equivalents (C♯/D♭, A♯/B♭, E♯/F, and so on). Assuming octave-invariance, the three members of a major triad have the relative semitone values n, $n+4$ and $n+7$; those of a minor triad n, $n+3$ and $n+7$.

During the 1970s and 1980s the music-psychologist Carol Krum-hansl conducted a ground-breaking series of experiments into the perception and cognition of musical pitch [Kru90]. By using the statistical technique of multi-dimensional scaling on the results of experiments on listeners' judge-

ments of inter-key relationships, she produced a table of coordinates in four-dimensional space which provides the basis for the lexical chord distance measure we adopt here. The 'distance' between triads a and b can be expressed as the four-dimensional Euclidean distance between these coordinates. We do not reproduce these distances here, but denote the distance as $Edist(a, b)$.

Now that these definitions are clear, we may proceed with the harmonic description algorithm. The basic idea is that when calculating the score of a simultaneity s on a lexical chord c, this score is influenced by all the other lexical chords p in which s participates. Thus, every lexical chord has an effect on every other lexical chord.

An analogy might help: The amount of gravitational force that two bodies (such as the earth and moon) exert on each other is proportional to the product of their masses, and inversely proportional to a function of the distance between them. By analogy, each of our 24 lexical chords is a body in space, and each exerts some influence on all others. Thus, if the notes of a G major triad are observed, not only does G major get the most mass, but we also assign some probability mass to E minor and B minor, a bit less to C major and D major, even less to A minor and F$^\sharp$ minor, and so on.

So the amount of influence exerted by each chord in the lexicon on the current chord is proportional to the number of pitches shared between the simultaneity s and each lexical chord p, and inversely proportional to the inter-triad distance from each p to c. Since, in general, 'contributions' of near neighbors in terms of inter-key distance are preferred, we use that fact as the basis for computing a suitable context:

$$Context(s, c) = \sum_{p \in lexicon} \frac{|s \cap p|}{Edist(p, c) + 1} \tag{A.2}$$

This context score is computed for every chord c in the lexicon (each point in the distribution), and then the entire distribution is normalized by the sum total of all context scores. While it is clear that the harmony of

all but the crudest music cannot be reduced to a mere succession of major and minor triads, as this choice of lexicon might be thought to assume, we believe that this is a sound basis for a probabilistic approach to harmonic description, as more complex chords (such as 7^{th} chords) are in fact accounted for by the contributions of their notes to the overall probabilistic context.

6.2 Smoothing

While the method above takes into account contributions from neighboring triads, it only does so within the current simultaneity, the current timestep. Harmony, as musicians perceive it, is a highly contextual phenomenon which depends not only on the harmonic distances at the current timestep, but is also influenced by the previous timesteps: the harmonies present in the recent past are assumed to be a good indication of the current harmony. Thus, a simultaneity with only one note might provide a relatively flat or uniform distribution across the lexical chord set, but when that simultaneity is taken in historical context, the distribution becomes more accurate.

We have developed a naive, yet effective, technique for taking into account this event-based context by examining a window of n simultaneities and using the values in that window to give a better estimate for the current simultaneity. This is given by the following equation, where s_t is the simultaneity at timestep t:

$$Smoothed(s_t, c) = \sum_{i=1}^{n} \frac{1}{i} \left(\sum_{p \in lexicon} \frac{|s_{t-i+1} \cap p|}{Edist(p, c) + 1} \right) \tag{A.3}$$

When the smoothing window n is equal to 1, this equation degenerates into the one from the previous section. When n is greater than one, the score for the lexical chord c at the current timestep is influenced by previous timesteps in proportion to the distance (number of events) between the current and previous timestep. As in the unsmoothed version, the smoothed context score is computed for every chord c in the lexicon and then the entire distribution is normalized by the sum total.

6.3 Markov Modeling

It should be clear by now that the primary difference between our harmonic description algorithm and most other such algorithms is the choice to create probabilistic *distributions* across the lexical chord set, rather than *reductions* of each simultaneity to a single, most salient lexical chord. The figure below is a toy example of a harmonic description, using an example lexicon of three chords, P, Q, and R. With this probabilistic harmonic description, we now create a Markov model.

	Timestep (Simultaneity)				
Lexical Chord	1	2	3	4	5
P	0.2	0.1	0.7	0.5	0
Q	0.5	0.1	0.1	0.5	0.1
R	0.3	0.8	0.2	0	0.9

Markov models are often used to capture statistical properties of a state sequence over time. We want to be able to predict future occurrences of a state by the presence of sequences of previous states. In our harmonic approach, we have chosen lexical chords as the states of the model. For an n^{th}-order model, a $24^n x 24$ matrix is constructed, with the 24^n rows representing the *previous state* space, and the 24 columns representing the *current state* space. An $(n+1)$ sized window slides over the sequence of lexical chord distributions and Markov chains are extracted from that window. The count of each chain is added to the matrix, where the cross of the first n states is the previous state, and the $(n+1)^{th}$ state is the current state. Finally, when the entire observable sequence has been counted, each row of the matrix is individually summed and the elements of each row normalized by the sum total for that row.

One problem is that Markov modeling only works on 1-dimensional sequences of observable states, while our harmonic description is a sequence of 24-point probability distributions. Our solution is to assume independence between points in each distribution at each timestep, so that an exhaustive

number of independent, one-dimensional paths through the sequence may be traced. (This exhaustive paths approach is abstractly similar to one suggested by Doraisamy and Rüger [DR01].) Each path, thus constructed, is not counted as a full observation. Instead, observations are proportional; the degree to which each path is observed is a function of the amount by which all elements of the path are present. Since independence between neighboring simultaneities was assumed, this becomes the product of the values of each state which comprises the path. For example, suppose we construct a 2^{nd}-order model from the sequence of distributions, above. Then one of the many observed state sequences we would see in timesteps 1 to 3 is "QRR". The count of this observation is 0.08 = (0.5 * 0.8 * 0.2).

7 Scoring Function

Our goal is to produce a ranked list for a query across the collection. We wish to rank those pieces of music at the top which are most similar to the query, and those pieces at the bottom which are least similar. This is the task of the scoring function. We have chosen as this function the Kullback-Liebler (KL) divergence, a measure of how different two distributions are, over the same event space. The divergence is always zero if two distributions are exactly the same, or a positive value if the distributions differ. We denote the KL divergence between query model q and music document model d as $D(q||d)$. "The KL divergence between [q] and [d] is the average number of bits that are wasted by encoding events from a distribution [q] with a code based on the not-quite-right distribution [d]" [MS01].

In our Markov model, each previous state, each row in the $24^n x24$ matrix, is a complete distribution. We therefore compute a divergence score for each row in the model, and add the value to the total divergence score for that query-document pair. This is given by the following equation, where q_i and d_i represent each previous state. It is imperative that the same modeling procedure and size that is used for the document models is also used for the query model.

Table A.1: **Average Ranks** for Transcription I

Bach Preludes			
	mm0	mm1	mm2
Window 1	4.83	23.11	219.41
Window 2	4.83	4.83	13.98
Window 3	4.76	3.52	4.30
Window 4	4.83	3.17	**3.04**
Random = 1575			

Bach Fugues			
	mm0	mm1	mm2
Window 1	4.04	35.08	192.08
Window 2	3.63	5.69	10.58
Window 3	3.31	5.19	3.52
Window 4	3.23	4.02	**2.38**
Random = 1575			

$$D(q||d) = \sum_{q_i \in q, d_i \in d} \left(\sum_{x \in X} q_i(x) \log \frac{q_i(x)}{d_i(x)} \right) \tag{A.4}$$

However, there is a problem in that sometimes a document model can have estimates of zero probability. This is especially true of shorter music documents, in which a lot of the possible transitions are never observed. The divergence score in such cases ($q_i(x) \log \frac{q_i(x)}{0}$)) automatically goes to infinity. This small problem in just a single value could therefore throw off our entire score for that document. We therefore must create some small but principled non-zero value for every document model zero value. There are many ways to do this, but we have done so by "backing off" to a general music model, using the value of that previous state node from the general model whenever we encounter a zero value in any particular document model.

A general music model is created by averaging the models over the entire set of document models in the collection. In principle, there could still

Table A.2: **Average Ranks** for Transcription II

Bach Preludes			
	mm0	mm1	mm2
Window 1	8.91	28.72	223.87
Window 2	7.85	5.04	16.72
Window 3	7.54	**3.83**	6.85
Window 4	7.35	4.87	7.96
Random = 1575			

Bach Fugues			
	mm0	mm1	mm2
Window 1	6.08	24.88	142.92
Window 2	5.33	4.77	10.23
Window 3	6.10	3.75	3.63
Window 4	5.79	3.58	**2.60**
Random = 1575			

remain zero values in the general music model, depending on the size and properties of the collection. In our experiments, however, we found this almost never to be the case. Also, it should be observed that when the query model has a zero probability in any cell, there is no problem. The KL divergence for that point is $0 \log \frac{0}{d_i(x)}$, which is zero.

8 Experiment Design and Results

For our retrieval experimentation, we adopt the Cranfield evaluation model[3] [CMK66]. This requires three crucial components: (1) Source collection, (2) Query, and (3) Relevance judgements which label each item in the source collection as either relevant or not relevant to the query. In all our experiments, the source collection remains the same. However, we vary the queries and the relevance judgements, as described below.

[3]See also http://ciir.cs.umass.edu/music2000/evaluation.html

8.1 Source Collection

The basic test collection on which we tested our retrieval method was assembled from data provided by the Center for Computer Assisted Research in the Humanities (CCARH) [CCA00]. It comprises around 3000 files of separate movements from polyphonic fully-encoded music scores by a number of classical composers (including Bach, Beethoven, Handel, and Mozart) of varying keys, textures (i.e. average numbers of notes in a simultaneity) and lengths (numbers of simultaneities). To this basic collection we add, for the purposes of the present paper, three additional sets of polyphonic music data, for a total collection of approximately 3,150 pieces of music. Collectively, we denote these Twinkle, Lachrimae and Folia variations as the TLF sets:

T 26 individual variations on the tune known to English speakers as 'Twinkle, twinkle, little star' (in fact a mixture of mostly polyphonic and a few monophonic versions);

L 75 versions of John Dowland's 'Lachrimae Pavan', collected as part of the ECOLM project (www.ecolm.org) from different 16th and 17th-century sources, sometimes varying in quality (numbers of 'wrong' notes, omissions and other inaccuracies), in scoring (for solo lute, keyboard or five-part instrumental ensemble), in sectional structure and in key;

F 50 variations by four different composers on the well-known baroque tune 'Les Folies d'Espagne'.

8.2 Experiment One: Known Item

The idea for the first experiment comes from a desire to test the robustness of our harmonic modeling. We therefore assembled from the Naxos audio collection the 24 Preludes and Fugues of Book I of Bach's Well-tempered Clavier. The score versions of these piano-based, human-played audio files are present within our source collection, from the CCARH data. So each audio-transcribed Prelude or Fugue becomes a query, and the score from

which the audio file was ostensibly played becomes the one "known item" relevant document in the collection.

The question is whether this degraded, transcribed query (Figure A.3) can retrieve, at a high rank relative to all other music in the collection, the original "perfect" score (Figure A.2). For this particular example, Figure A.2 was retrieved at a rank of 1^{st}, from our collection of 3,150 pieces of music.

As good as this result is, accurate evaluation deals with averages to get a true indication of system performance. The results of this experiment are found in Tables A.1 and A.2. For each set of queries (either the 24 Preludes or 24 Fugues) the known item was retrieved at some rank, where first is the best possible value. These ranks were then averaged across all queries in the set. Results are given for 0^{th} to 2^{nd}-order Markov models, each of which has been smoothed over a window of size $n = 1$ to $n = 4$. For comparison, a system which performed random ranking would place the known item, on average, approximately $1,575^{th}$.

Discussion: Our results show that the known item searches are extremely successful. Through a combination of higher-order Markov models and larger smoothing windows, we were able to retrieve the true symbolic version of the piece using the audio-transcribed, degraded query at an average rank of a little over 3 for the Bach Preludes, and a little over 2 for the Bach Fugues. While there is still room for improvement, it should prove difficult to produce an *average* which is better than 2nd or 3rd.

Though results vary slightly from the Transcription I to the Transcription II algorithms, equally good results were achieved using each. Our harmonic modeling technique is robust enough to handle two significantly different transcription algorithms.

8.3 Experiment Two: Variations

For the second experiment, we wish to determine whether our harmonic modeling approach is useful for retrieving variations on a piece of music, rather than just the original. Recall that in addition to the CCARH data,

our source collection contains three sets of variations. For this experiment, the audio version one variation is selected and the score versions of all the variations are judged "relevant" to the audio query, even though their actual *similarity* may vary considerably. A good retrieval system would therefore return all variations toward the top of the 3,150 item list, and all non-variations further down. This is repeated for all audio pieces in the set. For example, Figure A.4 contains a few of the "Twinkle" variations. When the audio version of Variation 3 is used as the query, we expect not only the score version of Variation 3 to be ranked highly, but the score version of Variation 11 and the score version of the Theme to be ranked highly as well. (The "Theme" is, of course, one of the many variations.)

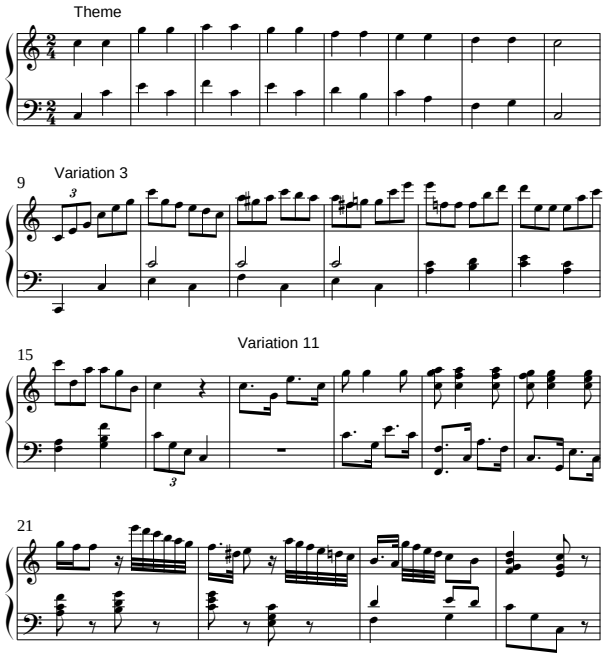

Figure A.4: Excerpts from the "Twinkle" variations

Because of the size of these sets and our limited resources, we were not able to get human performances of all these variations. Instead, we converted the queries to MIDI and used a high-quality (30 Megabyte) piano soundfont to create an audio "performance". This apparent weakness in our evaluation is countered by two facts: (1) These audio queries are still

polyphonic, even if synthesized, and automatic transcription of overlapping and irregular-duration tones is still quite difficult. (2) Many of the variations on a piece are themselves quite different from a potential query, as we see in Figure A.4, and good retrieval is still a difficult task. Even if the perfect score of a variation were used as a query, rather than the imperfect (though perhaps slightly better because of the synthesized audio), quality retrieval is not guaranteed. While we hope to work with a human-produced audio collection for this retrieval experiment someday, as we have done with the known-item Naxos data above, we feel the gist of the evaluation has not been compromised.

Table A.3: Variations Transcription I, Mean Average Precision

Twinkle			
	mm0	mm1	mm2
Window 1	0.164	0.130	0.168
Window 2	0.168	0.163	**0.179**
Window 3	0.168	0.122	0.131
Window 4	0.172	0.135	0.101
Random = 0.0052			

Lachrimae			
	mm0	mm1	mm2
Window 1	0.168	0.064	0.033
Window 2	0.168	0.140	0.094
Window 3	0.164	0.172	0.158
Window 4	0.162	0.179	**0.191**
Random = 0.0112			

Folia			
	mm0	mm1	mm2
Window 1	0.375	0.216	0.136
Window 2	0.379	0.365	0.219
Window 3	0.378	**0.479**	0.334
Window 4	0.384	0.445	0.390
Random = 0.0087			

Presentation of the known-item results were straightforward. With one relevant document in the entire collection, one need only report the rank (or average rank across all queries) of this document. The problem with multiple relevant documents is how best to visualize the ranked list. Typically this is done using 11-pt interpolated recall-precision graphs, with *precision* (number

Table A.4: Variations Transcription II, Mean Average Precision

Twinkle			
	mm0	mm1	mm2
Window 1	0.145	0.111	0.150
Window 2	0.145	0.149	**0.156**
Window 3	0.145	0.095	0.117
Window 4	0.130	0.104	0.083
Random = 0.0052			

Lachrimae			
	mm0	mm1	mm2
Window 1	0.172	0.056	0.030
Window 2	0.174	0.136	0.096
Window 3	0.173	0.177	0.162
Window 4	0.172	0.181	**0.195**
Random = 0.0112			

Folia			
	mm0	mm1	mm2
Window 1	0.333	0.172	0.105
Window 2	0.337	0.315	0.178
Window 3	0.331	**0.422**	0.284
Window 4	0.328	0.389	0.329
Random = 0.0087			

of relevant documents over total retrieved at a point in the ranked list) given at various level of *recall* (number of relevant documents retrieved over the total number of relevant documents in the query set). However, space constrains us. Instead, we present two values which hopefully characterize the data: mean average precision and mean precision at the top 5 retrieved documents.

Average precision is computed by calculating the precision for a single query (retrieved relevant over total retrieved) every time another variation (relevant document) is found, then averaging over all those points. This score is then averaged over all queries in the set, to create the mean average precision. It is a single value popular in Information Retrieval studies because it allows easy comparison of different systems.

However, some users are more interested in the precision of a system at the top of the ranked list. If the user does not care about finding every single variation but only cares about finding any variation, then the average

Table A.5: Variations Transcription I, Precision at top 5 retrieved pieces

Twinkle			
	mm0	mm1	mm2
Window 1	**0.592**	0.323	0.462
Window 2	0.577	0.500	0.515
Window 3	0.577	0.431	0.485
Window 4	0.585	0.485	0.415
Random = 0.0077			

Lachrimae			
	mm0	mm1	mm2
Window 1	0.496	0.067	0.056
Window 2	0.501	0.317	0.096
Window 3	0.477	0.520	0.451
Window 4	0.456	0.531	**0.616**
Random = 0.0213			

Folia			
	mm0	mm1	mm2
Window 1	0.692	0.104	0.212
Window 2	0.680	0.444	0.200
Window 3	0.704	**0.884**	0.544
Window 4	0.740	0.804	0.816
Random = 0.02			

precision is not as important as the precision at the top of the ranked list. We therefore compute the precision for a single query after retrieving the top 5 documents. If 1 of those documents is relevant (a variation), then the precision is 0.2, or 20%. If none of them are, the precision is 0%. If all of them are, the precision is 100%. We then average this value over all queries in the set, to get the mean precision at the top 5 retrieved documents.

Tables A.3 and A.4 contain the mean average precision results, while Tables A.5 and A.6 contain the average precision at the top 5 retrieved documents. These values are given for the three TLF query sets, for 0^{th} to 2^{nd}-order Markov models, each of which has been smoothed over a window of size $n = 1$ to $n = 4$, averaged over all queries in each of the TLF query sets. Unlike the known-item results, where the lower numbers were better because they represented average rank, the values for these variations experiments represent precision. Higher numbers are better.

For each query set we give, as a baseline, the expected value a random

Table A.6: Variations Transcription II, Precision at top 5 retrieved pieces

Twinkle			
	mm0	mm1	mm2
Window 1	0.485	0.285	0.408
Window 2	0.515	**0.539**	0.431
Window 3	0.531	0.346	0.446
Window 4	0.439	0.392	0.331
Random = 0.0077			

Lachrimae			
	mm0	mm1	mm2
Window 1	0.461	0.040	0.032
Window 2	0.440	0.216	0.059
Window 3	0.427	0.523	0.419
Window 4	0.440	0.499	**0.619**
Random = 0.0213			

Folia			
	mm0	mm1	mm2
Window 1	0.628	0.056	0.112
Window 2	0.672	0.404	0.144
Window 3	0.628	**0.788**	0.480
Window 4	0.608	0.728	0.732
Random = 0.02			

ranking algorithm would produce, for a document collection of size and with relevant document count equal to those of the various query sets. For example, the Twinkle set only has 26 variations, so a random ranking of the collection yields a mean precision at the top 5 documents of 0.0077. The Lachrimae set has 75 variations, so it is only natural that with more relevant documents in the collection, a random ranking of those documents will include more relevant documents toward the top of the list. Indeed, the mean precision at 5 docs of the random algorithm on the Lachrimae set is 0.0213.

Discussion: Using an audio-transcribed query to retrieve variations on a piece of music is a much harder problem. We do not consider this a solved problem by any means, but we are encouraged by the results we see. First, it is clear that our harmonic modeling algorithm is doing something correctly, as it yields significant improvement over the random algorithm. Second, we once again see the trend that higher order Markov models and

more harmonic smoothing yield better results. Higher and longer does not monotonically indicate better performance, but the trend is nonetheless apparent.

We also note that some query sets are more difficult than others. Not only did we have more success on the Folia variations than on the Twinkle variations, but after listening to the actual pieces, it is clear than human judges would have more difficulty picking out the Twinkle variations than they would the Folia variations. Nevertheless, even for these more difficult Twinkle variations, almost 3 of the 5 top ranked documents are, on average, relevant variations. We feel this is a respectable result.

9 Conclusion

It is now clear that retrieval of polyphonic scores using polyphonic audio is possible. By "taking apart" (transcribing) an audio music query and harmonically modeling the musically-salient pitch features we are bridging the gap between audio and symbolic music retrieval, and doing so within the difficult polyphonic domain.

That we have restricted ourselves in this paper to piano (a single timbre) is not a limitation as much as it is an indication of future potential. We did not have to perfectly recognize every single note in a piece of music in order for the harmonic modeling to be successful. Therefore, future audio transcription methods which attempt to transcribe the even more difficult polytimbral, polyphonic domain may do so with the confidence that the transcription need not be perfect in order to get good retrieval results.

The same technique which gives us robust, error-tolerant retrieval of known-item queries (Section 8.2) is also useful for retrieving variations (Section 8.3). Indeed, at one level of abstraction, a composed variation can be thought of as an "errorful transcription" of the original piece. Our harmonic modeling approach succeeded in capturing a degree of invariance, a degree of similarity, across such "transcriptions". The technique, though far from

perfect, is an important first step for polyphonic (audio and symbolic) music retrieval.

10 Future Work

We feel one useful direction for this work is to bypass the transcription phase and go directly from audio features to a harmonic description. This will make the modeling phase slightly more difficult, but there might be advantages to bypassing the transcription, as the transcription is only used to create harmonic descriptions. This would bring us closer to some harmonic-recognition work being carried out by others in the pure audio domain such as by Carreras et al [CLL99], or Fujishima [Fuj99].

A second direction is to modify the harmonic description smoothing algorithm. We propose in the future to adopt either a (millisecond) time-based or a (rhythmic) beat-based window smoothing approach, rather than the event-based approach we use in this paper. We will sum the harmonic contributions in the way described above across simultaneities within the window in inverse proportion to their time or beat-based distance from the current simultaneity, with additional weightings provided according to metrical stress, note duration or other factors that might be considered helpful. Indeed, harmonic smoothing, properly executed, might be a way of integrating the problematic, not-quite-orthogonal dimensions of pitch and duration within a polyphonic source. Better time-based smoothing might also yield a richer harmonic description, because it gives less weight to transient changes in harmony arising from non-harmonic notes such as passing tones or appoggiaturas.

A third direction deals with passage level retrieval. Rather than modeling entire documents, it might be useful to model portions of documents, particularly if those portions are musically salient. Finally, the issue of standardized test collections remains important. We are interested in participating in such experiments, to compare our system with others that will be developed in the future.

11 Acknowledgements

We would like to thank Eleanor Selfridge-Field, Craig Sapp, and Bret Aarden for their patient assistance with the CCARH data, which we used as our primary source collection. We would like to thank Naxos for the use of their Bach Prelude and Fugue audio recordings. Finally, Samer Abdallah deserves credit as an early inspiration for some of the harmonic description assumptions made in this paper.

Bibliography

[Abd02] Samer Abdallah. *Towards Music Perception by Redundancy Reduction and Unsupervised Learning in Probabilistic Models*. PhD thesis, King's College London, 2002.

[AHH+01] E. Allamanche, J. Herre, O. Hellmuth, B. Bernhard Fröbach, and M. Cremer. AudioID: Towards content-based identification of audio material. In *Proceedings of the 100th AES Convention, Amsterdam, The Netherlands*, May 2001.

[AKZ02] D. Arfib, F. Keiler, and U. Zölzer. *Digital Audio Effects*, chapter Time-frequency Processing. John Wiley and Sons, Ltd., 2002.

[AmH00] The American Heritage Dictionary of the English Language, fourth edition. Published by Houghton Mifflin Company, 2000.

[AMN01] AMNS. Nightingale music notation software. documentation at http://www.ngale.com., 2001.

[AP03] Samer Abdallah and Mark Plumbley. Probability as metadata: event detection in music using ICA as a conditional density model. Submitted to the 4^{th} International Symposium on Independent Component Analysis and Signal Separation, ICA2003, Nara, Japan, April 2003.

[Att59] F. Attneave. *Applications of Information Theory to Psychology, A Summary of Basic Concepts, Methods, and Results*. New York: Holt, Rinehart and Winston, 1959.

[BD85] J. Bloch and R. Dannenberg. Real-time accompaniment of polyphonic keyboard performance. In *Proceedings of the 1985 International Computer Music Conference, Vancouver*, pages 279–290, 1985.

[BDS02] J.P Bello, L. Daudet, and M. Sandler. Time-domain polyphonic transcription using self-generating databases. In *Proceedings of the 112th Convention of the Audio Engineering Society. Munich, Germany*, May 2002.

[BDW+01] W. Birmingham, R. B. Dannenberg, G. H. Wakefield, M. Bartsch, D. Bykowski, D. Mazzoni, C. Meek, M. Mellody, and W. Rand. Musart: Music retrieval via aural queries. *J. S. Downie and D. Bainbridge, editors, Proceedings of the 2nd Annual International Symposium on Music Information Retrieval (ISMIR), Indiana University, Bloomington, Indiana*, pages 73–81, October 2001.

[Bil93] Jeff Bilmes. Timing is of the essence: Perceptual and computational techniques for representing, learning, and reproducing expressive timing in percussive rhythm. Masters thesis, Massachusetts Institute of Technology, 1993.

[BMS00a] J.P Bello, G. Monti, and M. Sandler. An implementation of automatic transcription of monophonic music with a blackboard system. In *Proceedings of the Irish Signals and Systems Conference. Dublin, Ireland*, June 2000.

[BMS00b] J.P Bello, G. Monti, and M. Sandler. Techniques for automatic music transcription. In *Proceedings of the International Symposium on Music Information Retrieval. Plymouth, Massachusetts, USA*, October 2000.

[BPMS02] W. Birmingham, B. Pardo, C. Meek, and J. Shifrim. The Musart music-retrieval system. *D-Lib Magazine*, February 2002.

[Bre90] Albert S. Bregman. *Auditory Scene Analysis: The Perceptual Organization of Sound*. MIT Press, Cambridge, MA, 1990.

[Bro91] Judith Brown. Calculation of a constant Q spectral transform. *Journal of the Acoustic Society of America*, 89(1):425–434, January 1991.

[Bro93] Judith Brown. A high resolution fundamental frequency determination based on phase changes of the Fourier transform. *Journal of the Acoustic Society of America*, 94(2):662 – 667, November 1993.

[Bru57] J.S. Bruner. On perceptual readiness. *Psychological Review*, (64):123–152, 1957.

[BS00] J.P Bello and M. Sandler. Blackboard system and top-down processing for the transcription of simple polyphonic music. In *Proceedings of the COST G-6 Conference on Digital Audio Effects (DAFx-00). Verona, Italy*, December 2000.

[CCA00] CCARH. The musedata collection. http://www.musedata.org, 2000. Center for Computer Assisted Research in the Humanities, Stanford, CA.

[Cho97] Andrew Choi. Real-time fundamental frequency estimation by least-square fitting. *IEEE Transactions on Speech and Audio Processing,*, 5(2):201–205, March 1997.

[CJ86] C. Chafe and D. Jaffe. Source separation and note identification in polyphonic music. In *Proceedings of the International Conference on Acoustics, Speech, and Signal Processing (ICASSP), Tokyo*, volume 2, pages 1289–92. IEEE Press, 1986. Also available as Stanford University Department of Music Technical Report STAN-M-34, Palo Alto, CA.

[CJK+85] C. Chafe, D. Jaffe, K. Kashima, B. Mont-Rynaud, and J. Smith. Techniques for note identification in polyphonic music. In *Proceedings of the International Conference on Computer Music (ICMC), Burnaby, B.C., Canada.*, pages 399–406. Computer Music Association, 1985. Also available as Stanford University Department of Music Technical Report STAN-M-29, April 1986.

[CL91] N. Carver and V. Lesser. A new framework for sensor interpretation: Planning to resolve sources of uncertainty. In *The Proceedings of the 1991 National Conference on Artificial Intelligence (AAAI-91), Anaheim, California*, pages 725–731, July 1991.

[CLL99] F. Carreras, M. Leman, and M. Lesaffre. Automatic harmonic description of musical signals using schema- based chord decomposition. *Journal of New Music Research*, 28(4):310–333, 1999.

[CM80] T.A. Claasen and W.F. Mecklenbrauker. The Wigner distribution -A tool for time-frequency signal analysis- Part I: Continuous-time signals, Part II: Discrete-time signals, Part III: Relations with other time-frequency signal transformations. *Philips Journal of Research*, 35:217–250, 276–300 and 372–389, 1980.

[CMK66] C. W. Cleverdon, J. Mills, and M. Keen. *Factors Determining the Performance of Indexing Systems, Volume I - Design, Volume II - Test Results.* ASLIB Cranfield Project, Cranfield, 1966.

[CMR82] C. Chafe, B. Montreynaud, and L. Rush. Toward an intelligent editor of digital audio: Recognition of musical constructs. *Computer Music Journal*, 6(1):30–41, 1982.

[Com94] Pierre Comon. ICA - a new concept? *Signal Processing*, 36:287–314, 1994.

[Coo93] P. Cooke. *Modelling Auditory Processing and Organisation*. Cambridge University Press, Cambridge, 1993.

[CQFC94] F.J. Casajús-Quirós and P. Fernández-Cid. Real-time, loose-harmonic matching fundamental frequency estimation for musical signals. In *Proceedings of the 1994 IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP 1994)*, pages 221–224, 1994.

[CR83] R.E. Crochiere and L.R. Rabiner. *Multirate Digital Signal Processing*. Prentice Hall, 1983.

[Cro80] R.E. Crochiere. A weighted overlap-add method of short-time Fourier analysis/synthesis. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 281(1):99–102, 1980.

[Cyb84] G. Cybenko. Fast approximation of dominate harmonics. *SIAM J. Sci and State. Comp.*, 5:317–331, 1984.

[Dau90] Ingrid Daubechies. The wavelet transform, time-frequency localization and signal analysis. *IEEE Transactions on Information Theory*, 36(5):961–1005, September 1990.

[Dau01] Laurent Daudet. Transients modelling by pruned wavelet trees. In *Proceedings of the ICMC 2001 International Computer Music Conference, La Habana, Cuba*, 2001.

[DDS01] Chris Duxbury, Mike Davies, and Mark Sandler. Separation of transient information in musical audio using multiresolution analysis techniques. In *Proceedings of the COST G-6 Conference on Digital Audio Effects (DAFX-01), Limerick, Ireland*, December 2001.

[Dem77] A.P. Dempster. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society*, 39(1):1–38, 1977.

[Deu82] D. Deutsch. *The Psychology of Music*. New York: Academic Press, 1982.

[DGR93] P. Depalle, G. Garcia, and X. Rodet. Tracking of partials for additive sound synthesis using hidden Markov models. In *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing. ICASSP-93.*, volume 1, pages 225 –228, 1993.

[DH89] P. Desain and H. Honing. Quantization of musical time: A connectionist approach. *Computer Music Journal*, 13(3):56–66, 1989.

[Dix00a] Simon Dixon. Extraction of musical performance parameters from audio data. In *Proceedings of the 1st IEEE Pacific-Rim Conference on Multimedia (PCM 2000), Sidney, Australia*, 2000.

[Dix00b] Simon Dixon. On the computer recognition of solo piano music. In *Proceedings of the Australasian Computer Music Association Conference, Brisbane, Australia*, pages 31–37, 2000.

[Dol82] M.B. Dolson. *A tracking phase vocoder and its use in the analysis of ensemble sounds*. PhD thesis, California Institute of Technology, 1982.

[Dol86] Mark Dolson. The phase vocoder: A tutorial. *Computer Music Journal*, 10(4):14, 1986.

[Dov99] M. Dovey. An algorithm for locating polyphonic phrases within a polyphonic piece. In *Proceedings of AISB Symposium on Musical Creativity, Edinburgh*, pages 48–53, April 1999.

[DR01] S. Doraisamy and S. M. Rüger. An approach toward a polyphonic music retrieval system. In J. S. Downie and D. Bainbridge, editors, *Proceedings of the 2nd Annual International Symposium on Music Information Retrieval (ISMIR), Indiana University, Bloomington, Indiana*, pages 187–193, October 2001.

[EK90] M.W. Eysenck and M.T. Keane. *Cognitive psychology: a student's handbook*. Lawrence Erlbaum Associated Ltd., 1990.

[Ell96] D. P. W. Ellis. *Prediction-Driven Computational Auditory Scene Analysis*. PhD thesis, Department of Electrical Engineering and Computer Science, MIT, June 1996.

[EM88] R.S. Engelmore and A.J. Morgan. *Blackboard Systems*. Addison-Wesley publishing, 1988.

[Eva91] G. Evangelista. *Representations of Musical Sounds*, chapter Wavelet Transform that we can play. MIT Press, 1991.

[FCCQ98] P. Fernandez-Cid and F.J. Casajus-Quiros. Multi-pitch estimation for polyphonic musical signals. In *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, volume 6, pages 3565 – 3568, 1998.

[FG66] J.L. Flanagan and R.M. Golden. Phase vocoder. *Bell System Technical Journal*, (45):1493, 1966.

[FO96] D.J. Field and B.A. Olshausen. Emergence of simple-cell receptive properties by learing a sparse code for natural images. *Nature*, 381:607–609, 1996.

[Foo00] J. Foote. Arthur: Retrieving orchestral music by long-term structure. In *Proceedings of the 1st International Symposium for Music Information Retrieval (IS-*

MIR), Plymouth, Massachusetts, October 2000. See http://ciir.cs.umass.edu/music2000.

[FR90] P. Flandrin and O. Rioul. Affine smoothing of the Wigner-Ville distribution. In *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, volume 5, pages 2455 – 2458, 1990.

[FR91] N. Fletcher and T. Rossing. *The Physics of Musical Instruments*. Springer Verlag, 1991.

[FRD92] A. Freed, X. Rodet, and P. Depalle. Synthesis and control of hundreds of sinusoidal partials on a desktop computer without custom hardware. In *Int. Conf. on Signal Processing Applications and Technology (ICSPAT92)*, 1992.

[Fuj99] T. Fujishima. Realtime chord recognition of musical sound: a system using common LISP music. In *Proceedings of the 1999 International Computer Music Conference, Beijing, China*, pages 464–467, 1999.

[Gab46] D. Gabor. Theory of communication. *Journal of the Institution of Electrical Engineers*, 93(3):429–457, 1946.

[GB99] D. Godsmark and G. J. Brown. A blackboard architecture for computational auditory scene analysis. *Speech Communication*, 27:351–366, 1999.

[GBA00] A. De Gotzen, N. Bernardini, and D. Arfib. Traditional (?) implementations of a phase vocoder: the tricks of the trade. In *Proceedings of the COST G-6 Conference on Digital Audio Effects (DAFX-00), Verona, Italy*, December 2000.

[GBM+96] R. Gribonval, E. Bacry, S. Mallat, P. Depalle, and X. Rodet. Analysis of sound signals with high resolution matching pur-

suit. In *Proc. IEEE Symp. Time-Freq. and Time-Scale Anal. (TFTS'96)*, pages 125–128, June 1996.

[GH99] Masataka Goto and Satoru Hayamizu. A real-time music scene description system: Detecting melody and bass lines in audio signals. In *Working Notes of the IJCAI-99 Workshop on Computational Auditory Scene Analysis*, pages 31–40, 1999.

[Gib66] J.J. Gibson. *The senses considered as perceptual systems*. Boston: Houghton Mifflin, 1966.

[Gib79] J.J. Gibson. *The ecological approach to visual perception*. Boston: Houghton Mifflin, 1979.

[GLCS95] A. Ghias, J. Logan, D. Chamberlin, and B. Smith. Query by humming - musical information retrieval in an audio database. In *Proceedings of ACM International Multimedia Conference (ACMMM), San Francisco, CA*, pages 231–236, 1995.

[GM95] Masataka Goto and Yoichi Muraoka. Beat tracking based on multiple-agent architecture – A real-time beat tracking system for audio signals–. In Victor Lesser, editor, *Proceedings of the First International Conference on Multi–Agent Systems*. MIT Press, 1995.

[God97] S. Godsill. Bayesian enhancement of speech and audio signals which can be modelled as ARMA processes. *International Statistical Review*, 65:1–21, 1997.

[Goo96] M. Goodwin. Residual modeling in music analysis-synthesis. In *Proc. IEEE ICASSP*, pages 1005–1008, Atlanta, GA, 1996.

[Goo97] M. Goodwin. Matching pursuit with damped sinusoids. In *Proc. ICASSP-97, Munich, Germany*, 1997.

[Got00] Masataka Goto. A robust predominant-F0 estimation method for real-time detection of melody and bass lines in CD recordings. In *Proceedings of The 2000 IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP 2000)*, volume 2, pages 757–760, 2000.

[Got01] Masataka Goto. A predominant-F0 estimation method for CD recordings: MAP estimation using EM algorithm for adaptive tone models. In *Proceedings of the 2001 IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP 2001)*, pages 3365–3368, 2001.

[Gre72] R.L. Gregory. Seeing as thinking. *Times Literary Supplement*, June 1972.

[Gre80] R.L. Gregory. Perceptions as hypotheses. *Philosophical Transactions of the Royal Society of London*, (290):181–197, 1980. Series B.

[Hai01] Stephen Hainsworth. Analysis of musical audio for polyphonic transcription. First year report, Signal Processing Group, Department of Engineering, University of Cambridge, 2001.

[HAT96] K. N. Hamdy, A. Ali, and A. H. Tewfik. Low bit rate high quality audio coding with combined harmonic and wavelet representations. In *Proc. IEEE Intern. Conf. Acoust., Speech, and Sig. Processing (ICASSP)*, volume 2, pages 1045–1048, 1996.

[Haw93] Michael Hawley. *Structure out of Sound*. PhD thesis, MIT, 1993.

[Hec95] David Heckerman. A tutorial on learning with Bayesian networks. Technical Report MSR-TR-95-06, Microsoft Corporation, 1995.

[Hop82] John J. Hopfield. Neural networks and physical systems with emergent collective computational abilities. *Proceedings of the National Academy of Science*, 79:2554–2558, 1982.

[HRG01] H. H. Hoos, K. Renz, and M. Görg. Guido/MIR - an experimental music information retrieval system based on Guido music notation. In J. S. Downie and D. Bainbridge, editors, *Proceedings of the 2nd Annual International Symposium on Music Information Retrieval (ISMIR), Indiana University, Bloomington, Indiana*, pages 41–50, October 2001.

[HW01] Stephen W. Hainsworth and Patrick J. Wolfe. Time-frequency reassignment for musical analysis. In *Proeedings of the International Computer Music Conference, Havana, Cuba*, September 2001.

[Jeh97] Tristan Jehan. Musical signal parameter estimation. Msc thesis in electrical engineering and computer sciences, IFSIC, University of Rennes 1, France and Center for New Music and Audio Technologies, Berkeley, USA., 1997.

[JW92] J.Jeong and W.J. Williams. Kernel design for reduced interference distributions. *IEEE Transactions on Signal Processing*, 40(2):402 – 412, February 1992.

[KESV01] Anssi Klapuri, Antti Eronen, Jarno Seppänen, and Tuomas Virtanen. Automatic transcription of music. In *Symposium on Stochastic Modeling of Music, 14th Meeting of the FWO Research Society on Foundations of Music Research, Ghent, Belgium*, 2001.

[KH96] Kunio Kashino and Norihiro Hagita. A music scene analysis system with the MRF-based information integration scheme. In *Proceedings of the 13th Int. Conf. on Pattern Recognition (ICPR-96)*, volume II, pages 725–729, Aug 1996.

[KI89] H. Katayose and S. Inokuchi. The Kansei music system. *Computer Music Journal*, 13(4), 1989.

[Kla98a] A. Klapuri. Automatic transcription of music. Msc thesis, Department of Information Technology, Tampere University of Technology, April 1998.

[Kla98b] A. Klapuri. Number theoretical means of resolving a mixture of several harmonic sounds. In *Proceedings of the European Signal Processing Conference*, 1998.

[Kla99] A. Klapuri. Sound onset detection by applying psychoacoustic knowledge. In *IEEE International Conference on Acoustics, Speech and Signal Processing*, 1999.

[Kla01a] Anssi Klapuri. Means of integrating audio content analysis algorithms. In *Proceedings of the 110th Convention of the Audio Engineering Society, Amsterdam, The Netherlands*, 2001.

[Kla01b] Anssi Klapuri. Multipitch estimation and sound separation by the spectral smoothness principle. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing, Salt Lake City, USA, 2001.*, 2001.

[KM81] S.M. Kay and S.L. Marple. Spectrum analysis - A modern perspective. *Proceedings of the IEEE*, 69:1380–1419, 1981.

[KM88] R. Kronland-Martinet. The use of the wavelet transform for the analysis, synthesis and processing of speech and music sounds. *Computer Music Journal, MIT Press*, 12(4):11–20, December 1988.

[KM98] Kunio Kashino and Hiroshi Murase. Music recognition using note transition context. In *IEEE Int. Conf. on Acoustics, Speech and Signal Processing (ICASSP-98)*, volume VI, pages 3593–3596, May 1998.

[KNKT95] Kunio Kashino, Kazuhiro Nakadai, Tomoyoshi Kinoshita, and Hidehiko Tanaka. Organization of hierarchical perceptual sounds. In *Proceedings of the 14th Int. Joint Conf. on Artificial Intelligence (IJCAI-95), Vol.1*, pages 158–164, Aug 1995.

[KNKT98] K. Kashino, K. Nakadai, T. Kinoshita, and H. Tanaka. Application of Bayesian probability network to music scene analysis. In D. F. Rosenthal and H. G. Okuno, editors, *Compuational Auditory Scene Analysis*, NJ, May 1998. Lawrence Erlbaum Associates.

[Koh95] T. Kohonen. *Self-Organizing Maps*. Berlin/Heidelberg: Springer-Verlag, 1995.

[Kru90] C. L. Krumhansl. *Cognitive Foundations of Musical Pitch*. Oxford University Press, New York, 1990.

[Kun84] Milan Kundera. *The unbearable lightness of being*. Faber and Faber, 1984.

[KVES01] Anssi Klapuri, Tuomas Virtanen, Antti Eronen, and Jarno Seppänen. Automatic transcription of musical recordings. In *Consistent and Reliable Acoustic Cues Workshop, CRAC-01, Aalborg, Denmark*, 2001.

[KVH00] Anssi Klapuri, Tuomas Virtanen, and Jan-Markus Holm. Robust multipitch estimation for the analysis and manipulation of polyphonic musical signals. In *Proceedings of the COST-G6 Conference on Digital Audio Effects (DAFx-00) Verona, Italy*, 2000.

[KZC98] R. Keren, Y.Y. Zeevi, and D. Chazan. Automatic transcription of polyphonic music using the multiresolution Fourier transform. In *Proceedings of the Mediterranean Electrotechnical Conference, 1998. MELECON 98*, volume 1, pages 654 –657, 1998.

[LE77] V.R. Lesser and L.D. Erman. A retrospective view of the Heasay-II architecture. In *Proceedings of the Fifth International Joint Conference on Artificial Intelligence (IJCAI-77)*, pages 790–800, 1977.

[Lep99] P. Lepain. Polyphonic pitch extraction from musical signals. *Journal of New Music Research*, 28(4):296–309, 1999.

[Lev98] Scott Levine. *Audio Representations for Data Compression and Compressed Domain Processing*. PhD thesis, Stanford University, 1998.

[LK94] E.W. Large and J.F. Kolen. Resonance and the perception of musical meter. *Connection Science*, 6(1), 1994.

[LNGK93] V. Lesser, S.H. Nawab, I. Gallastegi, and F. Klassner. IPUS: An architecture for integrated signal processing and signal interpretation in complex environments. In *Proceedings of the 11th National Conference on Artificial Intelligence*, pages 249–255, 1993.

[LNK95] V. R. Lesser, S. H. Nawab, and F. I. Klassner. IPUS: An architecture for the integrated processing and understanding of signals. *Artificial Intelligence*, 77(1):129–171, Aug 1995.

[LPA91] P.J. Loughlin, J.W. Pitton, and L.E. Atlas. New properties to alleviate interference in time-frequency representations. In *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, volume 5, pages 3205 – 3208, 1991.

[LPA92] P.J. Loughlin, J.W. Pitton, and L.E. Atlas. An information-theoretic approach to positive time-frequency distributions. In *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, volume 5, pages 125 – 128, 1992.

[LPA93] P.J. Loughlin, J.W. Pitton, and L.E. Atlas. Bilinear time-frequency representations: new insights and properties. *IEEE Transactions on Signal Processing*, 41(2):750 –767, Feb. 1993.

[LT00] K. Lemström and J. Tarhio. Searching monophonic patterns within polyphonic sources. In *Proceedings of the RIAO Conference, College of France, Paris*, volume 2, pages 1261–1278, April 2000.

[Mah89] R. C. Maher. *An Approach for the Separation of Voices in Composite Musical Signals*. PhD thesis, University of Illinois, Urbana-Champaign, 1989.

[Mah90] R. C. Maher. Evaluation of a method for separating digitized duet signals. *Journal of the Audio Engineering Society*, 38(12):956–979, Dec 1990.

[Mar87] S.L. Marple. *Digital spectral analysis with applications*. Prentice Hall, Englewood Cliffs, NJ, 1987.

[Mar89] S.L. Marple. A tutorial overview of modern spectral estimation. In *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, volume 4, pages 2152 –2157, 1989.

[Mar96a] K. D. Martin. Automatic transcription of simple polyphonic music: Robust front end processing. In *Presented at the Third Joint Meeting of the Acoustical Societies of America and Japan, December*, 1996. Also available as M.I.T. Media Lab Perceptual Computing Technical Report no. 399, November 1996.

[Mar96b] K. D. Martin. A blackboard system for automatic transcription of simple polyphonic music. Technical Report 385, MIT Media Lab, Perceptual Computing Section, July 1996. Available at ftp://sound.media.mit.edu/pub/Papers/kdm-TR385.ps.gz.

[Mar98] Matija Marolt. Feedforward neural networks for piano music transcription. In *Proceedings of XII Colloqium on Musical Informatics, Gorizia, Italy.*, 1998.

[Mar99] Matija Marolt. A comparison of feed forward neural network architectures for piano music transcription. In *Proceedings of the International Computer Music Conference (ICMC 1999), Beijing, China*, pages 314–317, 1999.

[Mar00a] Matija Marolt. Adaptive oscillator networks for partial tracking and piano music transcription. In *Proceedings of the 2000 International Computer Music Conference, Berlin, Germany*, 2000.

[Mar00b] Matija Marolt. Transcription of polyphonic piano music with neural networks. In *Information technology and electrotechnology for the Mediterranean countries. Vol. 2, Signal and image processing : Proceedings of the 10th Mediterranean Electrotechnical Conference, MEleCon 2000, Cyprus*, volume 2, pages 29–31, 2000.

[Mar01] M. Marolt. SONIC : transcription of polyphonic piano music with neural networks. In *Proceedings of Workshop on Current Research Directions in Computer Music, Barcelona, November 15-17, 2001*, 2001.

[Mas96] P. Masri. *Computer Modeling of Sound for Transformation and Synthesis of Musical Signal*. PhD thesis, University of Bristol, 1996.

[MD92] C.M. McIntyre and D.A. Dermott. A new fine-frequency estimation algorithm based on parabolic regression. In *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing. ICASSP-92.*, volume 2, pages 541 –544, 1992.

[Mel91] D. K. Mellinger. *Event Formation and Separation in Musical Sound*. PhD thesis, Center for Computer Research in Music and Acoustics, Stanford University, 1991. Also Dept of Music Report STAN-M-77.

[Mer02] Merrian-Webster's Collegiate Dictionary, tenth edition. Published by Merrian-Webster Inc., 2002.

[MF02] Luis Gustavo P.M. Martins and Anibal J.S. Ferreira. PCM to MIDI transposition. In *Proceedings of the 112th Convention of the Audio Engineering Society, Munich, Germany*, 2002.

[MGB97] B. Moore, B. Glasberg, and T. Bear. A model for the prediction of thresholds, loudness and partial loudness. *Journal of the Audio Engineering Society*, 45(4):224–239, 1997.

[MGT98] B. Mulgrew, P. Grant, and J. Thompson. *Digital Signal Processing: Concepts and Applications*. Palgrave Macmillan Ltd, 1998.

[MH91] R. Meddis and M. Hewitt. Virtual pitch and phase sensitivity of a computer model of the auditory periphery. I: Pitch identification. *Journal of the Acoustical Society of America*, 89(6):2866–2882, 1991.

[Moo75] J. A. Moorer. *On the Segmentation and Analysis of Continuous Musical Sound by Digital Computer*. PhD thesis, Dept. of Computer Science, Stanford University, 1975. Available as Stanford University Department of Music Technical Report STAN-M-3.

[Moo77] J. A. Moorer. On the transcription of musical sound by computer. *Computer Music Journal*, 1(4):32–38, 1977.

[MQ86] R.J. McAulay and T.F. Quatieri. Speech analysis/synthesis based on a sinusoidal representation. *IEEE Transactions in*

Acoustics, Speech and Signal Processing, ASSP-34:744–754, 1986.

[MR97] Dirk Moelants and Christian Rampazzo. *KANSEI - The Technology of Emotion*, chapter A Computer System for the Automatic Detection of Perceptual Onsets in a Musical Signal, pages 141–146. Genova: AIMI-DIST, 1997.

[MS01] C. D. Manning and H. Schütze. *Foundations of Statistical Natural Language Processing*. MIT Press, 2001.

[MS02] G. Monti and M. Sandler. Pitch locking monophonic music analysis. In *Proceedings of the 112th Convention of the Audio Engineering Society (AES), Munich, Germany, May 10-13*, 2002.

[MSBW97] R. J. McNab, L. A. Smith, D. Bainbridge, and I. H. Witten. The New Zealand digital library melody index. *D-Lib Magazine*, May 1997. Available at: www.dlib.org/dlib/may97/meldex/05witten.html.

[MWL01] D. Meredith, G. Wiggins, and K. Lemström. Pattern induction and matching in polyphonic music and other multi-dimensional datasets. In *the 5th World Multi-Conference on Systemics, Cybernetics and Informatics, Orlando*, pages 61–66, 2001.

[MWT80] W. Marslen-Wilson and L.K. Tyler. The temporal structure of spoken language understanding. *Cognition*, (8):1–71, 1980.

[MZ93] S. Mallat and Z. Zhang. Matching pursuits with time-frequency dictionaries. *IEEE Transactions on Signal Processing*, 41:3397–3415, 1993.

[Nei67] U. Neisser. *Cognitive psychology*. New York: Appleton-Century-Crofts, 1967.

[Nei76] U. Neisser. *Cognition and Reality.* San Francisco: W.H. Freeman, 1976.

[New62] A. Newell. Some problems of the basic organization in problem-solving programs. In M.C. Yovits, G.T. Jacobi, and G.D. Goldstein, editors, *Proceedings of the second conference on self-organizing systems*, pages 393–423. Spartan Books, 1962.

[NI86] T. Niihara and S. Inokuchi. Transcription of sung song. In *Proceedings of the International Conference on Acoustics, Speech, and Signal Processing*, pages 1277–1280, 1986.

[Nii86a] H.P. Nii. Blackboard systems (Part 1). *AI Magazine*, 7(2):38–53, 1986.

[Nii86b] H.P. Nii. Blackboard systems (Part 2). *AI Magazine*, 7(3):82–106, 1986.

[NMB01] H. Neuchsmied, H. Mayer, and E. Battle. Identification of audio titles on the internet. In *Proceedings of the International Conference on Web delivering of music, Florence, Italy*, November 2001.

[OBCQ02] Luis I. Ortiz-Berenguer and F.J. Casajús-Quirós. Pattern recognition of piano chords based on physical model. In *Proceedings of the 112th Convention of the Audio Engineering Society, Munich, Germany*, 2002.

[OW89] F. Opolko and J. Warpnick. McGill University master samples (MUMS), Faculty of music, McGill University, Montreal, Canada. CD-ROM set, 1989.

[OW99] J.C. O'Neill and W.J. Williams. Shift covariant time-frequency distributions of discrete signals. *IEEE Transactions on Signal Processing*, 47(1):133 – 146, January 1999.

[PAB+01] M. D. Plumbley, S. A. Abdallah, J. P. Bello, M. E. Davies, J. Klingseisen, G. Monti, and M. B. Sandler. ICA and related models applied to audio analysis and separation. In *Proceedings of the Fourth International ICSC Symposium on Soft Computing and Intelligent Systems for Industry, Paisley, Scotland*, 2001.

[Par97] Amos Paran. Bottom-up and top-down processing. *English Teaching Professional*, (3), April 1997.

[PB98] Miller Puckette and Judith Brown. Accuracy of frequency estimates using the phase vocoder. *IEEE Transactions on Speech and Audio Processing*, 6(2):166–176, March 1998.

[PBO00] H. Purwins, B. Blankertz, and K. Obermayer. A new method for tracking modulations in tonal music in audio data format. citeseer.nj.nec.com/purwins00new.html, 2000.

[PC02] J. Pickens and T. Crawford. Harmonic models for polyphonic music retrieval. In *Proceedings of the ACM Conference in Information Knowledge and Management (CIKM), McLean, Virginia*, November 2002.

[Pea86] J. Pearl. Fusion, propagation, and structuring in belief networks. *Artificial Intelligence*, 29(3):241–288, 1986.

[Pea91] E.R.S. Pearson. *The Multiresolution Fourier Transform and its Application to Polyphonic Audio Analysis*. Phd thesis, University of Warwick, UK, September 1991.

[PG77] M. Piszczalski and B. A. Galler. Automatic music transcription. *Computer Music Journal*, 1(4):24–31, 1977.

[PG99] Douglas Preis and Voula Chris Georgopoulos. Wigner distribution representation and analysis of audio signals: An illus-

trated tutorial review. *Journal of the Audio Engineering Society*, 47(12):1043 – 1053, December 1999.

[Pic00] J. Pickens. A comparison of language modeling and probabilistic text information retrieval approaches to monophonic music retrieval. In *Proceedings of the 1st International Symposium for Music Information Retrieval (ISMIR)*, October 2000. See http://ciir.cs.umass.edu/music2000.

[Pon98] J. M. Ponte. *A Language Modeling Approach to Information Retrieval*. Phd thesis, University of Massachusetts Amherst, 1998.

[Por76] M.R. Portnoff. Implementation of the digital phase vocoder using the fast Fourier transform. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 24(3):243–248, 1976.

[PW96] W.J. Pielemeier and G.H. Wakefield. A high-resolution time-frequency representation for musical instrument signals. *Journal of the Acoustical Society of America*, 99(4):2382–2396, Apr 1996.

[PWS96] W.J. Pielemeier, G.H. Wakefield, and M.H. Simoni. Time-frequency analysis of musical signals. In *Proceedings of the IEEE*, volume 84, pages 1216 –1230, Sept. 1996.

[RB01] W. Rand and W. Birmingham. Statistical analysis in music information retrieval. In J. S. Downie and D. Bainbridge, editors, *Proceedings of the 2nd Annual International Symposium on Music Information Retrieval (ISMIR), Indiana University, Bloomington, Indiana*, pages 25–26, October 2001.

[REFN73] D.R. Reddy, L.D. Erman, R.D. Fennel, and R.B. Neely. The Hearsay speech understanding system: an example of the recognition process. In *Proceedings of the third international joint*

conference on artificial intelligence (IJCAI-73), pages 185–93, 1973.

[RGL96] L. Rossi, G. Girolami, and L. Leca. Spectral identification of polyphonic piano signals. *Acustica*, 82:S187–S187, Jan-feb 1996.

[RGL97] L. Rossi, G. Girolami, and M. Leca. Identification of polyphonic piano signals. *Acustica*, 83(6):1077–1084, Nov-dec 1997.

[RHW86] David E. Rumelhart, Geoffrey E. Hinton, and Ronald J. Williams. *Parallel Distributed Processing*, volume 1, chapter Learning Internal Representations by Error Propagation, pages 318–362. The MIT Press, 1986.

[RJ01] Xavier Rodet and Florent Jaillet. Detection and modeling of fast attack transients. In *Proceedings of the International Computer Music Conference*, 2001.

[Rod97] X. Rodet. Musical sound signal analysis/synthesis: Sinusoidal + residual and elementary waveform models. In *Proc. IEEE Time-Frequency and Time-Scale Workshop (TFTS97)*, 1997.

[Ros92] D. Rosenthal. *Machine Rhythm: Computer Emulation of Human Rhythm Perception.* PhD thesis, Department of Computer Science, MIT, 1992.

[Sch85] A. W. Schloss. *On the Automatic Transcription of Percussive Music - From Acoustic Signal to High-Level Analysis.* PhD thesis, Department of Hearing and Speech, Stanford University, 1985. Available as Stanford University Department of Music Technical Report STAN-M-27.

[Sch98] E. D. Scheirer. Tempo and beat analysis of acoustic musical signals. *J. Acoust. Soc. Am.*, 103(1):588–601, Jan 1998.

[Sen85] S. Seneff. *Pitch and Spectral Analysis of Speech based on an Auditory Synchrony Model.* Phd thesis, M.I.T., 1985.

[Ser89] X. Serra. *A System for Sound Analysis / Transformation / Synthesis Based on a Deterministic plus Stochastic Decomposition.* PhD Diss., Stanford University, 1989.

[Shu96] Tim Shuttleworth. *A Multiresolution Approach to the Transcription of Polyphonic Musical Signals using Neural Networks.* Phd thesis, The University of Warwick, 1996.

[SL94] J. Settel and C. Lippe. Real-time musical applications using the FFT-based resynthesis. In *Proceedings of the International Computer Music Conference (ICMC94)*, 1994.

[Sla93] M. Slaney. An efficient implementation of the Patterson-Holdsworth auditory filter bank. Technical Report 35, Apple Computer Co., 1993.

[SSW99] A. Sterian, M. H. Simoni, and G. H. Wakefield. Model-based musical transcription. In *Proceedings of the International Computer Music Conference (ICMC-99), Beijing, China*, pages 460–463, 1999. Also at http://musen.engin.umich.edu/papers/transcription.pdf.

[Sta83] J.P. Stautner. Analysis and synthesis of music using the auditory transform. M.s. thesis in electrical engineering and computer science, M.I.T., May 1983.

[SW92] H. Scott and R.G. Wilson. A comparison of filters for audio signal segmentation in audio restoration. Research Report RR231, Department of Computer Science, University of Warwick, UK, October 1992.

[SW95a] Tim Shuttleworth and Roland Wilson. A neural network for triad classification. In *Proceedings of the International Computer Music Conference, Banff, Canada*, 1995.

[SW95b] Tim Shuttleworth and Roland Wilson. The recognition of musical structures using neural networks. In *Working notes of the Workshop on Artificial Intelligence and Music, International Joint Conference on Artificial Intelligence, Montreal, Canada*, pages 101–105, 1995.

[SW97] A. Sterian and G.H. Wakefield. A frequency-dependent bilinear time-frequency distribution for improved event detection. In *Proceedings of the International Computer Music Conference, Thessaloniki, Greece*, 1997.

[SYHC+01] I. Shmulevich, O. Yli-Harja, E. Coyle, D. Povel, and K. Lemström. Perceptual issues in music pattern recognition - complexity of rhythm and key find. *Computers and the Humanities*, 35(1):23–35, 2001. Appeared also in the Proceedings of the AISB'99 Symposium on Musical Creativity.

[Tan93] Andranick S. Tanguiane. *Artificial Intelligence and Music Recognition*. Number 746 in Lecture Notes in Artificial Intelligence. Springer-Verlag, 1993.

[TEC01] G. Tzanetakis, G. Essl, and P. Cook. Automatic musical genre classification of audio signals. In J. S. Downie and D. Bainbridge, editors, *Proceedings of the 2nd Annual International Symposium on Music Information Retrieval (ISMIR), Bloomington, Indiana, Indiana University*, pages 205–210, October 2001.

[TG00] Harvey Thornburg and Fabien Gouyon. A flexible analysis-synthesis method for transients. In *Proc. International Computer Music Conference ICMC-2000*, Berlin, 2000.

[UZ99] A. Uitdenbogerd and J. Zobel. Melodic matching techniques for large music databases. In *Proceedings of ACM Interna-*

tional Multimedia Conference (ACMMM), Orlando, Florida, USA. ACM Press, October 1999.

[vB83] A. von Brandt. Detecting and estimating parameter jumps using ladder algorithms and likelihood ratio test. In *Proc. ICASSP*, pages 1017–1020, Boston, MA., 1983.

[Vil48] J. Ville. Théorie et application de la notion de signal analytique. *Cables et Transmissions*, 2A(1):61–74, 1948.

[vIM92] L. van Immerseel and J.P. Martens. Pitch and voiced/unvoiced determination with an auditory model. *Journal of the Acoustical Society of America*, 91(6):3511–3526, 1992.

[VLM97] T. Verma, S. Levine, and T. Meng. Transient modeling synthesis: a flexible analysis/synthesis tool for transient signals. In *Proc. of the International Computer Music Conference*, Greece, 1997.

[vS00] Thomas von Schröeter. Auto-regressive spectral line analysis of piano tones. Technical Report 7, Department of Computing. Imperial College of Science, Technology, and Medicine, 2000.

[vSDR00] Thomas von Schröeter, Shyamala Doraisamy, and Stefan M. Rüger. A road map from raw polyphonic audio to locating recurring themes. In *Proceedings of the first International Symposium in Music Information Retrieval (ISMIR 2000), Plymouth, Mass., USA*, 2000.

[War70] R. M. Warren. Perceptual restoration of missing speech sounds. *Science*, (167):392–393, 1970.

[WCPD92] R.G. Wilson, A.D. Calway, E.R.S. Pearson, and A.R. Davies. An introduction to the multiresolution Fourier transform and its applications. Research Report RR204, Department of Computer Science, University of Warwick, UK, January 1992.

[Wei98] Eric W. Weisstein. *The CRC Concise Encyclopedia of Mathematics*. CRC Press, 1998.

[WGR99a] P. J. Walmsley, S. J. Godsill, and P. J. W. Rayner. Bayesian graphical models for polyphonic pitch tracking. In *Proc. Diderot Forum on Mathematics and Music, Vienna, Austria*, 2–4 December 1999. Also at http://www-sigproc.eng.cam.ac.uk/p̃jw42/ftp/didlt.pdf.

[WGR99b] P. J. Walmsley, S. J. Godsill, and P. J. W. Rayner. Bayesian modelling of harmonic signals for polyphonic music tracking. In P. J. Walmsley and P. Wolfe, editors, *Cambridge Music Processing Colloquium*, Cambridge, UK, 30 September 1999.

[WGR99c] P. J. Walmsley, S. J. Godsill, and P. J. W. Rayner. Polyphonic pitch tracking using joint Bayesian estimation of multiple frame parameters. In *Proceedings of the 1999 IEEE Workshop on Applications of Signal Processing to Audio and Acoustics, New Palz, New York*, 17–20 October 1999.

[WW70] R. M. Warren and R. P. Warren. Auditory illusions and confusions. *Scientific American*, (223):30–36, 1970.

[ZAM90] Y. Zhao, L.E. Atlas, and R.J. Marks. The use of cone-shaped kernels for generalized time-frequency representations of non-stationary signals. *IEEE Transactions on Acoustics, Speech and Signal Processing*, 38(7):1084 –1091, July 1990.