

1 Introduction

Transcription of music is defined to be the act of listening to a piece of music and of writing down musical notation for the notes that constitute the piece [Martin96a]. In other terms, this means transforming an acoustic signal into a symbolic representation, which comprises *notes*, their *itches* (see Table 1), *timings*, and a classification of the *instruments* used. It should be noted that musical practice does not write down the *loudnesses* of separate notes, but determines them by overall performance instructions.

A person without a musical education is usually not able to transcribe polyphonic music, in which several sounds are playing simultaneously. The richer is the polyphonic complexity of a musical composition, the more experience is needed in the musical style and the instruments in question, and in music theory. However, skilled musicians are able to resolve even rich polyphonies with such a flexibility in regard to the variety of instrumental sounds and musical styles that automatic (computational) transcription systems fall clearly behind humans in performance.

The automatic transcription of *monophonic* signals is practically a solved problem since several algorithms have been proposed that are reliable, commercially applicable and operate in real time. Attempts towards *polyphonic* transcription date back to the 1970s, when Moorer built a system for transcribing duets, i.e., two-voice compositions [Moorer75b]. His system suffered from severe limitations on the allowable frequency relations of two simultaneously playing notes, and on the range of notes in use. However, this was the first polyphonic transcription system, and several others were to follow.

Until these days transcription systems have not been able to solve other than toy problems. It is only the two latest proposals that can transcribe music 1) with more than two-voice polyphony, 2) with even a limited generality of sounds, and 3) yield results that are mostly reliable and would work as a helping tool for a musician [Martin96b, Kashino95]. Still, these two also suffer from substantial limitations. The first was simulated using only a short musical excerpt, and could yield good results only after a careful tuning of threshold parameters. The second system restricts the range of note pitches in simulations to less than 20 different notes. However, some flexibility in transcription has been attained, when compared to the previous systems.

Potential applications of an automatic transcription system are numerous. They include tools that musicians and composers could use to efficiently analyze compositions that they only have in the form of acoustic recordings. A symbolic representation of an acoustic signal allows flexible mixing, editing, and selective coding of the signal. Automatic transcription would facilitate a music psychological analysis of performed music, not to speak about the help it would provide in maintaining acoustic archives and their statistical analysis. Some people would certainly appreciate a radio receiver capable of tracking jazz music simultaneously on all stations.

As can be seen, applications of a music transcription system may be compared to those of a *speech recognition* system. Both are very complicated tasks, but the massive commercial interests in speech coding and recognition have attracted much more attention, in contrast with the relatively little amount of research on music recognition. This is why the authors of transcrip-

tion systems have often been single enthusiastic individuals in research groups that also make manifold audio research on topics that are more readily commercially applicable.

The automatic transcription of music is related to several branches of science. A fundamental source of knowledge is found in *psychoacoustics*, which studies the perception of sound, including, e.g., speech and music. *Computational auditory scene analysis* is an area of research that has a somewhat wider scope than that of ours. It sets out to analyse the acoustic information coming from a physical environment and to interpret the numerous distinct events in it [Ellis96]. *The analysis of musical instrument sounds* supports music transcription since music consists of the sounds of various kinds of musical instruments and of human voices. *Digital signal processing* is a branch of information science that, along with psychoacoustics, is of fundamental importance for us. It is concerned with the digital representations of signals and the use of computers to analyse, modify, or extract information from signals [Ifeachor93].

In Chapter 2, we describe a literature review on automatic music transcription and on the related areas of interest. In Chapter 3, we discuss a decomposition of the transcription problem into more tractable pieces, and present some approaches that have been taken. Chapter 4 is devoted to rhythm tracking (defined in Table 1) and Chapter 5 to methods that can be used in tracking the fundamental frequency of sounds. *The most important chapters are 6 and 7, since they constitute the original part of this thesis.* In Chapter 6, we develop a number theoretical method of observing sounds in polyphonic signals. The performance of the algorithm will be evaluated when we apply it to an automatic transcription of piano music in Chapter 7. Finally, in Chapter 8 we discuss perception-based primitive dependencies in music and so-called top-down processing, which utilizes internal, high-level models and predictions in transcription.

Table 1: Term definitions

fundamental frequency	See <i>pitch</i> .
loudness	<i>Perceived</i> attribute of a sound that corresponds to the physical measure of sound <i>intensity</i> , the degree of being loud. Loudness is a psychological description of the magnitude of an auditory sensation [Moore95].
pitch	<i>Perceived</i> attribute of sounds, which enables an ordering of sounds on a scale extending from low to high. <i>Fundamental frequency</i> is a corresponding physical (not perceptual) term. We use pitch and fundamental frequency as synonyms.
note	We use the word <i>note</i> in two roles: to refer to a symbol used in musical notations and to refer to a sound that is produced by a musical instrument, when that symbol is played.
rhythm tracking	1) Finding the times when separate sounds start and stop playing in an acoustic signal. 2) Giving a description of the rhythmic structure of a musical piece, when the timings of separate sounds have been given. <i>A rhythm tracking system</i> may comprise only the latter or both.

2 Literature Review

“The interesting jobs must also be done by someone.”

-Esa Hämäläinen, physics student at TUT

A literature review was conducted on the automatic transcription of music and related areas of interest. These are psychoacoustics, computational auditory scene analysis, musical instrument analysis, rhythm tracking, fundamental frequency tracking, and digital signal processing. The meaning of these terms was given in the introduction.

In this chapter, we first describe the methods that were used in search of the literature. Then we briefly describe the state-of-the-art and history of designing automatic transcription systems. Finally, we summarize and discuss the most important references we found in the abovementioned research areas.

2.1 Methods

The research in the Signal Processing Laboratory at Tampere University of Technology comprises several areas of interest, including audio signal processing, but an analysis of musical signals had not been attempted before this project. Therefore, the first step in finding music transcription literature was to contact the Acoustic Laboratory of Helsinki University of Technology. The staff of that department provided us with initial references and a wider framework of the research topics that should be involved.

After becoming familiar with the relevant topics, we started searching for publications on them. A major facilitating observation was that most universities of technology give their publication indices on-line on the *World Wide Web*. In addition to that, we conducted searches in the proceedings of international signal processing *conferences*, and in *Ei Compendex*, which is the most comprehensive interdisciplinary engineering information database in the world. Keywords that we used in the search operations were the topics of the research areas and their principal concepts. The most important *international journals* in the search were Journal of the Acoustic Society America, International Computer Music Journal, IEEE Transactions on Acoustics, Speech and Signal Processing, and Journal of the Audio Engineering Society.

We studied again and in detail the publication indices of the most promising research institutes, which were Machine Listening Group of the Massachusetts Institute of Technology, Center for Computer Research in Music and Acoustics of the University of Stanford, and Institut de Recherche et Coordination Acoustique / Musique in Paris. After having publications we studied their reference lists. In the very final phase we followed reference chains and, when necessary, corresponded with the authors via e-mail. This resulted in a collection that we consider covering enough.

Our scope and treatise is limited by several factors, but especially by the limited amount of resources compared to the wide range of topics that are related with music transcription. Moreover, engaging in a research that is quite new to our laboratory, analysis of musical signals, called for paying the required attention to just finding the right points of emphasis and avoiding wrong assumptions in an early phase. *Obtaining* the publications was easier than we expected -

we failed to have only a very few publications.

2.2 Published transcription systems

The state-of-the-art in music transcription will be discussed in coming chapters, where the sub-problems of the task are taken under consideration. There we will also refer to the most up-to-date research in different areas. In this section we will take a glance at the different transcription systems that have been presented until now.

We summarize some figures of merit of the different systems in Table 3. These performance statistics were not explicitly stated in some publications, but had to be deduced from the presented simulation material and results. For this reason, the figures should be taken as rough characterizations only. Furthermore, it is always hard to know how selective the presentation of simulation results of each system has been, and how much has been attained just by a careful tuning of parameters. In the table, *polyphony* refers to the maximum polyphony in presented transcription simulations, *sounds* represent the instruments that were involved, *note range* gives the number of different note pitches involved, and *knowledge used* column lists the types of knowledge that were incorporated into each system. System *architectures*, such as a straightforward abstraction hierarchy or a blackboard, will be discussed in Chapter 8.

Table 2: Transcription systems

Reference	Institute	Performance	Knowledge used
Moorer75a,b	Stanford University	<i>Polyphony</i> : 2 (severe limitations on content). <i>Sounds</i> : violin, guitar. <i>Note range</i> : 24.	Heuristic approach.
Chafe82,85,86	Stanford University	<i>Polyphony</i> : 2 (presented simulation results insufficient). <i>Sound</i> : piano. <i>Note range</i> : 19.	Heuristic approach.
Maher89,90	Illinois University	<i>Polyphony</i> : 2. <i>Sounds</i> : clarinet, bassoon, trumpet, tuba, synthesized. <i>Note ranges</i> : severe limitation, pitch ranges must not overlap.	Heuristic approach.
Katayose89	Osaka University	<i>Polyphony</i> : 5 (several errors allowed). <i>Sounds</i> : piano, guitar, shamisen. <i>Note r.</i> : 32.	Heuristic approach.
Nunn94	Durham University	<i>Polyphony</i> : up to 8 (several errors allowed, perceptual similarity). <i>Sound</i> : organ. <i>Note range</i> : 48.	Perceptual rules. <i>Architecture</i> : bottom-up abstraction hierarchy.
Kashino93,95	Tokyo University	<i>Polyphony</i> : 3 (quite reliable). <i>Sounds</i> : flute, piano, trumpet, automatic adaptation to tone. <i>Note range</i> : 18.	Perceptual rules, timbre models, tone memories, statistical chord transition dictionary. <i>Architecture</i> : blackboard, Bayesian probability network.
Martin96a,b	MIT	<i>Polyphony</i> : 4 (quite reliable). <i>Sound</i> : piano. <i>Note range</i> : 33.	Perceptual rules. <i>Architecture</i> : blackboard

The first polyphonic transcription system, that of Moorer's, was introduced in Chapter 1 [Moorer75b]. Moorer's work was carried on by a *group of researchers at Stanford* in the beginning of the 1980s [Chafe82,85,86]. Further development was made by Maher [Maher89,90]. However, polyphony was still restricted to two voices, and the range of fundamental frequencies for each voice was restricted to nonoverlapping ranges.

In the late 1980s, *Osaka University in Japan* started a project which was aimed at extracting sentiments (feelings) from musical signals, and at constructing a robotic system that could respond to music as a human listener does [Katayose89]. Two transcription systems were designed in the course of the project. One of them transcribed monophonic Japanese folk-songs, and employed knowledge of the scale in Japanese songs to cope with the ambiguity of the human voice. The other transcribed polyphonic compositions for piano, guitar, or shamisen. The polyphony of this system was extended up to *five simultaneous voices*, but only at the expense of allowing some more errors to occur in the output.

In 1993, *Hawley* published his research on computational auditory scene analysis, and also addressed the problem of transcribing polyphonic piano compositions [Hawley93]. We failed to have his publication, but according to Martin [Martin96b], Hawley's system was reported to be fairly successful.

Douglas Nunn works with transcription at Durham University, UK. His transcription system is characterized by a mainly heuristic signal processing approach, and has been applied to synthetic signals which involve even up to eight simultaneous organ voices [Nunn94]. However, Nunn emphasizes perceived similarity between the original and transcribed pieces, allowing a few more errors to occur in the output.

A significant stride was taken in the history of automatic transcription, when a group of researchers at the *University of Tokyo* published their transcription system, which employed several new techniques [Kashino93]. They were the first to clearly list and take into use human auditory separation rules, i.e., auditory cues that promote either fusion or segregation of simultaneous frequency components in a signal. Further, they introduced tone model based processing (using information about instrument sounds in processing), and proposed an algorithm for automatic tone modeling, which means automatic extraction of the tone models from the analysed signal. In 1995, they further improved the system by employing a so-called blackboard architecture, which seems to be particularly fitted to transcription since it allows a flexible integration of information from several diverse knowledge sources without a global control module [Kashino95]. The architecture was used to implement a Bayesian probability network, which propagates the impact of new information through the system.

Another recent transcription system, that of *Keith Martin's* (MIT), also uses a blackboard architecture [Martin96a]. He has put quite a lot of effort in implementing the blackboard structure, but does not utilize high-level musical knowledge to the same extent as that of Kashino's, and does not build a probabilistic information propagation network. Automatic tone modeling is not addressed, either. However, along with Kashino's system, Martin's approach represents the state-of-the-art in music transcription. Later, Martin still upgraded his system by adding a perceptually more motivated front end, which employs correlograms (discussed in Chapter 3) in signal analysis [Martin96b].

2.3 Related work

The discussion above concerned implemented transcription systems that are purported to transcribe polyphonic music consisting of harmonic sounds (no drums). As stated earlier, there are several other fields of science that are related to music transcription. Here we introduce some of the most important research.

Two excellent sources of information in the field of psychoacoustics are [Bregman90] and

[Moore95]. *Albert Bregman's* book "Auditory Scene Analysis - the Perceptual Organization of Sound" (773 pages) comprises results of a three decade research work of this experimental psychologist, and has been widely referenced in the branches of computer science that are related to auditory perception. Music perception is also addressed in the book. Another, newer and not so well known, is "Hearing - Handbook of Perception and Cognition" (468 pages), which is edited by *Brian Moore*, and also covers research on auditory perception over times. Both of these are excellent sources of psychoacoustic information for the design of an automatic transcription system.

Computational auditory scene analysis (CASA) refers to the computational analysis of the acoustic information coming from a physical environment, and the interpretation of numerous distinct events in it. In 1991, *David Mellinger* prepared a review of psychoacoustic and neuropsychological studies concerning the human auditory scene analysis [Mellinger91]. He did not implement a complete computer model of the auditory system, but tested them computationally, actually using musical signals as test material. More recently, the work of *Daniel Ellis* represents the up to date research on CASA [Ellis96]. His study also comprises prediction-driven processing, which means utilization of the predictions of an internal world model and higher-level knowledge. Ellis evaluated his computational model, and obtained a good agreement between the events detected by the model and by human listeners.

Our research on the analysis of musical instrumental sounds was limited by time, and mainly covers the sinusoidal representation, which we found the most useful. This will be discussed in Section 3.3. Some references on that area are [McAulay86, Smith87, Serra89,97, Qian97]. Meillier's comment on the importance of the attack transient of a sound is worth noticing [Meillier91].

Some systems have been proposed that are aimed at transcribing polyphonic music which consists of drum-like instruments only [Stautner82, Schloss85]. Since they are more related to the systems that track rhythm, we will discuss them in Chapter 4. Table 4 in that chapter summarizes the rhythm tracking systems. Monophonic transcription, more generally called *fundamental frequency tracking*, will be treated in Chapter 5.

2.4 Roadmap to the most important references

In Figure 1 we summarize the most important references of this thesis (see References for a complete list). Researchers' names are arranged on horizontal and vertical axes into geographical and institutional groups, and the space in between represents the reference relations of the researches.

Several disclaimers must be stated. First of all, the interrelations have been constructed from the material which we used in writing this thesis and which was found using the literature review method explained earlier. Some researchers may be well aware of each other, although it does not appear in our material. Second, the research interests, such as 'rhythm tracking', only represent the topics that we utilized, definitely not characterizing all their work. Third, the institutional position may not correspond to the recent situation, but reflects the position at the time of writing the referred material. Fourth, the selection of researchers to this map from among all references is according to our judgment.

Despite these disclaimers, some structure is brought to the pile of literature that literally lies on our desk. The more there are dots on a *horizontal* line of a researcher, the more he/she has been

referred to, and the more relevant his/her work is likely to be from the point of view of those who refer. Reading the *vertical* lines tells how much a researcher has referred to the others, and how much he/she has studied the same material as we. *Blocks of crossings* between two institutes give some cues about how much the institutes utilize each other's work.

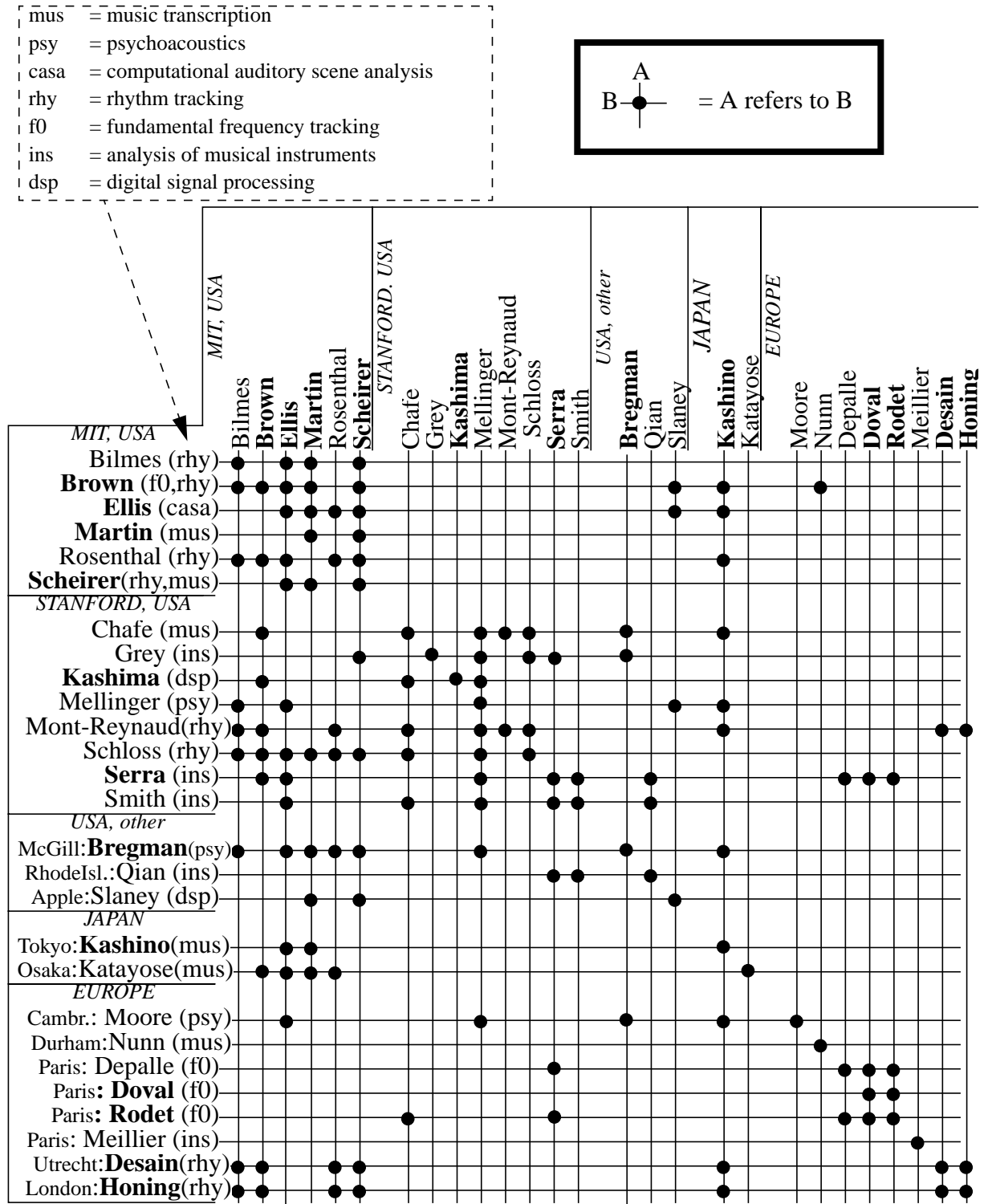


Figure 1. Interrelationships of the most important references of this thesis. Key references are bolded. See the explanations and disclaimers in the text.

2.5 Commercial products

Not even the first commercial transcription system has been released which would be able to transcribe polyphonic music. On the contrary, monophonic transcription machines have been integrated to several studio equipment. They include pitch-to-MIDI changers and allow symbolic editing and fixing of mistuned singing, for example [Opcode96]. The acronym MIDI stands for Musical Instrument Digital Interface, and is a standard way of representing and communicating musical notes and their parameters between two digital devices [General91].

3 Decomposition of the Transcription Problem

Comparing the approaches of different transcription systems reveals a significant overall similarity in decomposing the transcription problem into more tangible pieces and in the way information is represented at successive stages of abstraction from an acoustic signal to note symbols.

In this chapter we discuss potential approaches, abstraction hierarchies and selections of concepts in a transcription system. This is needed to reveal underlying potentially wrong assumptions in our approach, and to avoid overlooking approaches that do not seem to be promising at a first glance. We first discuss the relevance of the *note* as a target symbol, and then list the basic components of a transcription system.

Most of this chapter is devoted to finding an appropriate *representation for information at mid-level* between the acoustic signal and its musical notation. The term mid-level is used to refer to the level of processing in auditory perception which occurs between an acoustic low-level signal reaching the ear and its cognitive high-level representation [Ellis95].

At the end of the chapter, we present a design philosophy that was formed in the course of studying these matters.

3.1 Relevance of the note as a representation symbol

Scheirer remarks on two implicit assumptions in most transcription systems: a unidirect *bottom-up* flow of information from low-level processes to high-level processes, and the use of *note* as a fundamental representational symbol in music perception [Scheirer96a]. The first of these, a pure bottom-up flow of data, does not apply to the most recent transcription systems [Kashino93,95, Martin96a,b]. Top-down processing will be defined and treated in Chapter 8. The second question, whether a note is an appropriate symbol in representation or not, will now be discussed to a certain extent.

We cannot assume notes being the fundamental mental representation of all musical perception, or there being a transcription facility in brains for that [Scheirer96a]. Experimental evidence indicates that, instead of notes, humans extract *auditory cues* that are then grouped into percepts. Predictive models that utilize musical knowledge and context are used in grouping.

Bregman pays attention to the fact that music often wants the listener to accept simultaneous sounds as a single coherent sound with its own striking emergent properties. The sound is *chimeric* in the sense that it does not belong to any single physical source. The human auditory system has a tendency to segregate a sound mixture to the physical sources, but orchestration is often called upon to oppose these tendencies and force the auditory system to create a single chimeric sound, which would be irreducible into perceptually smaller units [Bregman90]. This is a problem in music transcription, as will be seen when an attempt is made to resolve polyphonic musical signals in Chapter 6.

In this thesis we do not try to understand the mental processes in human music perception. *Our intention is to transcribe acoustic musical signals into a symbolic representation which musi-*

cians could use to reproduce the acoustic signal using a limited variety of musical instrument sounds. When chimeric sounds are in question, they must be decomposed to musical instrument sounds in order to be reproducible.

The symbol note expresses the fundamental frequency of a sound that should be played using a certain musical instrument. Thus, it is most appropriate and relevant as far as we require that a selected symbolic representation should enable transforming it back to an acoustic signal using those instruments. Additional symbols, such as the loudness of the sounds and the rhythmic structure of a musical composition must be added, but the symbol note is fundamental among them.

3.2 Components of a transcription system

Figure 2 illustrates the basic components of a transcription system and their interrelations. The different facets of transcription, rhythm tracking, multipitch tracking and top-down processing, will be discussed in indicated chapters of this thesis.

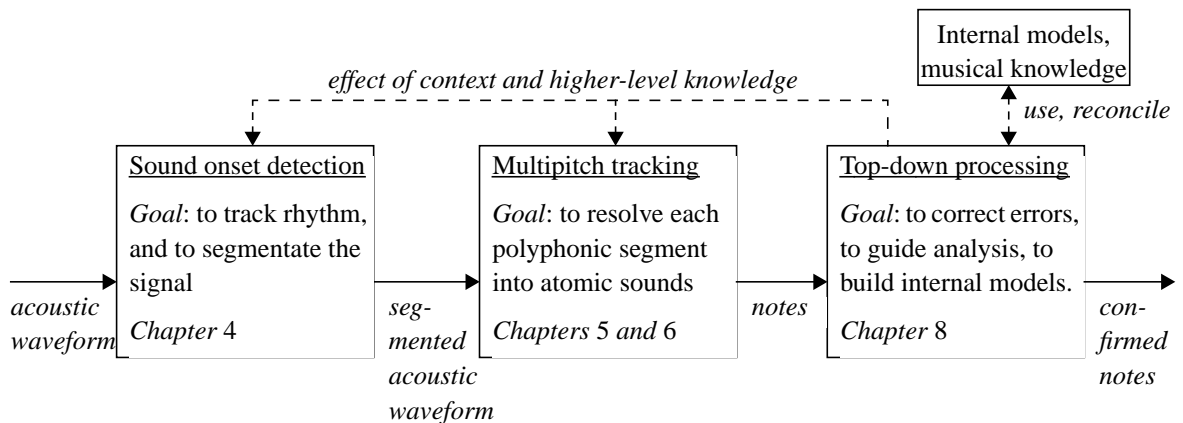


Figure 2. Basic components of a transcription system and their interrelations. Overall goal of each and the corresponding chapters of this thesis are indicated.

The transcription system that we design and simulate does not implement top-down processing, although that topic will be extensively discussed in Chapter 8. Thus the simulation results will be presented in Chapter 7, right after the first two components have been designed, and before taking top-down processing into consideration.

Multipitch tracking is the most involved component, since the path from an acoustic signal to notes is long and complicated. It also plays a crucial role in determining the general approach of a transcription system. Therefore in the rest of this chapter, we focus on discussing a suitable chain of information representations on the path to yield notes.

3.3 Mid-level representation

Auditory perception may be viewed as a hierarchy of representations from ‘low’ to ‘high’, where low-level representations are appropriate to describe an acoustic signal reaching the ear, and high-level representations are those to which we have cognitive access, such as a comprehended sentence of a language. Knowledge of the physiological and psychological processes at *mid-level* between these two is most restricted.

We consider the waveform of an acoustic signal to be a low-level representation, and a musical

notation (notes) to be a high-level representation. Between these two, intermediate abstraction level(s) are indispensable, since the symbols of a musical notation cannot be straightly deduced from the information that is present in the acoustic signal. Mid-level representations of several transcription systems were reviewed, and are listed in Table 3. As stated earlier, this reveals a fundamental resemblance in the signal analysis of the different systems.

Table 3: Transcription systems and their mid-level representations

Reference	Mid-level representation
Moorer75	filterbank, periodicities
Pisczalski77,81	local maxima in frequency domain
Chafe82,85,86	local maxima in frequency domain
Niihara86	sinusoid tracks
Katayose89	sinusoid tracks
Nunn92	sinusoid tracks
Kashino93, 95	sinusoid tracks
Martin96a	sinusoid tracks
Martin96b	correlogram

In his doctoral thesis, Daniel Ellis makes an extensive study on computational auditory scene analysis [Ellis96]. He has also published a paper comprising an excellent analysis and comparison of the mid-level representations [Ellis95]. Ellis classifies different representations according to three conceptual ‘axes’:

- the choice between a fixed and a variable bandwidth in frequency analysis
- discreteness, the degree to which the representation is structured to meaningful chunks
- dimensionality of the transform - some representations possess an extra dimension in addition to the usual pair of time and frequency

Using these axes, Ellis places the different representations to the corners of a cube, which acts as a ‘representation space’ (Figure 3). The benefits of each representation are now discussed.

The general requirement of a mid-level representation is that it may be computed efficiently from the input, and that it can readily answer the questions asked of it by the higher levels of processing. *Fast Fourier Transform (FFT)* can be efficiently calculated, but is not satisfying, since the human ear has a logarithmic frequency resolution and has the ability of trading between time and frequency resolutions, which is not possible with a fixed bandwidth. From

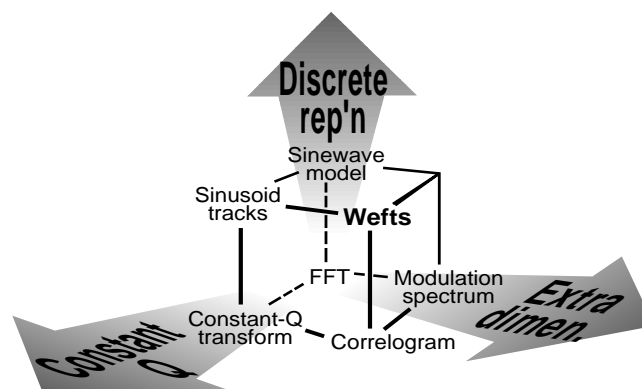


Figure 3. Three dimensions for the properties of sound representations define a cube, upon which representations may be placed [Ellis95, reprinted by permission].

the music point of view, frequencies that make up the scale of Western music are logarithmically spaced.

Thus we move along the ‘variable bandwidth’ axis to *constant Q transform (CQT)*, which provides logarithmic frequency resolution. The transform is explained in [Brown88], and an efficient algorithm for calculating it is given in [Brown92]. A fundamental shortcoming of FFT is removed, but CQT fails to fulfil a certain criterion given for a mid-level representation: it neither reduces the number of objects in representation nor increases the meaning of each object.

Both FFT and constant Q transform have been used in monophonic fundamental frequency tracking [Brown91b], but it is only by moving along the two remaining axes that we arrive at the two representations that have been used in recent polyphonic transcription systems. We acquire a discrete representation by tracking contiguous regions of local energy maxima in time-frequency, called *sinusoid tracks*. This representation is particularly successful in making a discrete representation of the signal, since resynthesizing the sound from the sinusoid tracks produces a result that possesses a high degree of perceptual fidelity to the original, in spite of being a poor approximation in mean-squared error sense [Ellis95, Serra97]. This representation will be further described and motivated in Section 3.4, since it is our selection among the mid-level representations.

Moving along the third axis of Ellis’s cube, to the direction of adding extra dimensions to a representation, is motivated by some psychoacoustic observations. Patterson proposes a perception model, which leads to a mid-level representation called *correlogram* [Patterson92, Slaney93]. A correlogram is calculated by applying a short-time autocorrelation to the outputs of a constant Q frequency filter bank (typically 20-40 frequency bands). This will result in a three-dimensional volume, where the dimensions are time, frequency, and lag time of short-time autocorrelations applied to the outputs of filter bank filters.

Practically a correlogram means searching for *periodicities* from the outputs of a filter bank. Common periodicities found at different frequency bands can be further associated to compose a more discrete representation, to which Ellis gives the name *weft*. Weft is a contiguous periodicity in time-frequency-periodicity volume, having a certain length in time and extending over the frequency bands that share the same periodicity. Weft representation fulfils another criterion for a good mid-level representation: it tries to organize sounds to their independent sources. Actually wefts, as defined by Ellis, track periodicities in the amplitude envelope of the signal, not in the signal itself. A straightforward discretization of a correlogram is sometimes called a summary autocorrelogram or *periodogram*.

Although quite complicated, periodograms and wefts are motivated by the fact that they explain a wide range of psychoacoustic phenomena and oddities, and have proved to be practically efficient in computational auditory scene analysis. A publicly available implementation of Patterson’s model has been released by Slaney [Slaney93].

Only one transcription system uses the correlogram as its mid-level representation [Martin96b]. A certain analogy with correlograms can also be seen in some rhythm tracking algorithms, where a bank of filters is followed by a network of resonators to detect the strongest periodicities [Scheirer96b]. Rhythm tracking will be discussed in Chapter 4.

3.4 Reasons for rejecting the correlogram and choosing sinusoid tracks

We studied carefully the psychoacoustic motivations of the correlogram and its practical

appropriateness to solve concrete transcription problems. We also made simulations using Slaney’s implementation of the auditory filterbank and correlogram [Slaney93]. We came into the conclusion that the correlogram and wefts suit computational auditory scene analysis well in general, but are not very appropriate to the transcription of polyphonic music. This result is definitely not evident, and is therefore justified in the following.

The problem with the correlogram and weft analysis

Let us repeat what was said earlier about chimeric sounds: “music often wants the listener to accept simultaneous sounds as a single coherent sound with its own striking properties”. We want to emphasize that this is an important principle in music in general, not only at single particular points. A certain tendency in Western music is to pay attention to the *frequency relations* of simultaneously played sounds. Two sounds are in a *harmonic* relation to each other, if their frequencies are in rational number relations. In Section 6.3 we will show that the closer is the relation, the more perfectly the notes play together and the more chimeric is their mixture.

Even the most typical mixtures of three notes may be so chimeric that they cannot be resolved using a straightforward periodicity tracking algorithm. We pick one example from a wider set that will be given in Section 6.3: In a basic major chord, the two lower notes, *C* and *E*, overlap 60 percent of the frequency partials of a third note, *G*. This means that sixty percent of the partials of the note *G* would be found from the signal even in the absence of the note *G*, and in that case a straightforward periodicity tracker will easily be misled to find a ‘ghost’ *G*.

In Section 3.1 we reviewed Scheirer’s problematization of the note being a relevant representational symbol in music perception. What was not emphasized there is that, indeed, the note is not a representational symbol for music *perception*. It was accepted especially because of our intention to produce a symbolic representation that could be used by musicians to reproduce an acoustic signal. *We suggest that human perception model as such is not appropriate for the transcription of music.* An average person is not able to transcribe even the mixtures of only a few musical sounds, even if the individual sounds had been clearly introduced before posing the transcription problem. Since music transcription is not a ‘natural’ perceptual task, but can be seen as separating chimeric sounds, we suggest that specific methods are needed instead of a general computational auditory scene analysis.

Both correlograms and weft analysis apply a short-time autocorrelation to the outputs of a certain auditory filter bank. The filter bank is not intended to provide a sufficiently precise frequency resolution, but it is the subsequent autocorrelation that is used for tracking periodicities in the signal. Utilization of autocorrelation is a problem here, since *autocorrelation fuses information on perceptual grounds in such a way that it prevents a separate treatment of each harmonic partial* that we consider necessary in order to resolve musical polyphonies. The correlogram and wefts were motivated by their plausibility from a perceptual point of view, but it should be noted that these representations are so involved that they also tie the subsequent algorithms to themselves.

Advantages of sinusoid tracks

A weakness of sinusoid track representation is that it remains at a lower abstraction level, not assigning sinusoids to their due sources of production. However, this can also be seen as an advantage: the representation is very compact, but different pieces of information are still separated from each other. This is important since in musical signals there is no straightforward

way of assigning sinusoidals to their sources of production, so that it could be done already at mid-level. Keeping the sinusoids distinct allows the higher levels to apply specific algorithms that take into account the particular characteristics of musical signals.

Another advantage of sinusoid tracks is their high invertibility: the representation is discrete and compact, but resynthesizing the sound from its sinusoid tracks still produces a result that possesses a high perceptual fidelity to the original. Sinusoid track representation is in accordance with recent research on sound modeling, where sinusoid models (sometimes called additive synthesis) are considered most appropriate [McAulay86, Smith87, Serra89,97, Qian97]. These models do not presume the sound to be harmonic, although it helps. Noisy components can be separately modelled if needed [Serra89,97].

Sinusoids are tracked in the frequency domain, since separate sinusoids cannot be tracked without making a frequency transform. As reminded earlier, autocorrelation tracks periodicities, not single sinusoids. Some time domain pitch detection algorithms will be discussed later in Chapter 5.

3.5 Extracting sinusoid tracks in the frequency domain

Frequency transforms

The human ear is characterized by a logarithmic frequency resolution, i.e., the absolute resolution is better at the low end. This is why the frequencies that have been chosen to make up the scale of Western music are on a logarithmic scale, too. A *fast Fourier transform (FFT)* yields components that are separated by a constant frequency interval. Thus FFT does not map to musical frequencies efficiently, despite its other benefits.

The solution is a frequency transform, in which the frequency bins are geometrically spaced, i.e., resolution is constant in the *logarithmic* frequency scale. This is called *constant Q transform (CQT)*, since it yields a constant ratio between the center frequency and a resolution of each frequency domain bin. The transform is explained in [Brown88] and an efficient implementation via the fast Fourier transform is proposed in [Brown92]. However, the efficient implementation of CQT relies on FFT in such a way that it practically combines several bins of FFT to produce a logarithmic frequency resolution. It therefore fails to interchange the weaker frequency resolution at the high end to a better time resolution which would be desirable to model human hearing. On the other hand, straightforward calculation of CQT is too laborious to be practically useful.

Bounded Q transform (BQT) is a hybrid technique that makes efficient use of FFT to produce a constant number of bins *per each octave* [Kashima85]. BQT is calculated as follows. FFT is calculated and half of the frequency bins are discarded, storing only the top octave. Then the time domain signal is low pass filtered and downsampled with a factor of two, and FFT is calculated with the same time window length, which now gives twice the previous resolution. From this transform, the top half of bins is again stored, which now corresponds to the second highest octave and comprises an equal number of frequency bins with the highest octave. This procedure is repeated until the lowest octave of interest is reached. The advantage of this method is that it practically has the speed of the FFT, with variable frequency and time resolution and is thus able to *optimize both time and frequency resolution* [Brown88].

The results of BQT can still be refined so that the resolution of each individual bin is the same

in a logarithmic scale. This can be done by applying Brown’s algorithm for an efficient calculation of constant Q transform to the results of BQT. However, since this practically just loses some of the information of BQT, it should not be done, if an exactly logarithmic resolution or visual inspection of sounds is not wanted [Brown91b]. We concluded sinusoidal tracks to be an appropriate mid-level representation, and thus it is enough to efficiently extract the time-frequency information to be used in tracking sinusoids. This is achieved by bounded Q transform.

A musical sound often consists of an *attack transient* followed by a steady sound. This structure is due to a short-lived interaction of an external force, which drives the sound playing. In Section 8.4 we will discuss the significance of the attack transient, and consider high-resolution spectral estimation methods that are needed to observe it.

Building sinusoid tracks in the frequency domain

Sinusoids appear as local energy maxima, peaks, in the frequency domain. Their evolution in time can be traced by a series of frequency transforms at successive time points. There is a multitude of approaches towards spectral estimation and tracking of sinusoids [McAulay86, Smith87, Serra89,97, Qian97, Depalle93, Ding97]. Some of them are presented in the context of a transcription system [Katayose89, Kashino93]. At present, our transcription system does not utilize the time evolution of the sinusoids, but only their instantaneous amplitudes. Thus, we cannot give a comparison of these methods that would be justified by simulation results.

3.6 Selected approach and design philosophy

In Section 3.2 we illustrated the basic components of a transcription system and remarked that decisions in the multipitch tracking module play a crucial role in defining the approach of a system. The selected representations in multipitch tracking are summarized in Figure 4, where they are placed in the earlier presented overview.

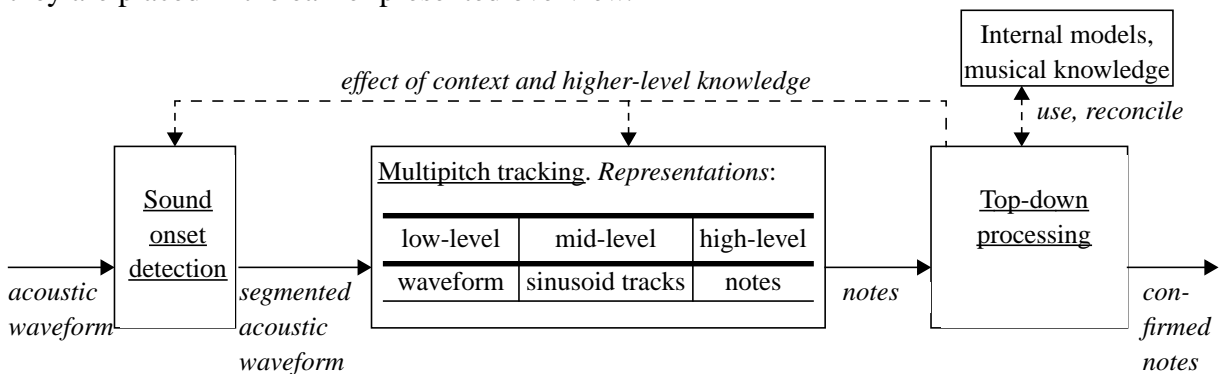


Figure 4. Selected representations at different abstraction levels, and their position in our transcription system that will be described later.

It should be noted that a hierarchy of representations does not imply a unidirectional flow of information from low-level processes to the higher levels. Mid-level representations may be introduced, and the effect of high-level internal models and predictions still propagates down through the representational layers. Therefore, data *representations* and processing *algorithms* should be discussed separately, according to the desirable properties of each. Top-down processing will be discussed in Chapter 8.

The distinction between onset detection and multipitch tracking is not evident. Rhythm track-

ing might also take place together with the sinusoid tracking, in which case the onset times are deduced from the onsets of the sinusoids. Figures 2 and 4 just depict our approach.

A certain design philosophy was formed in the course of studying the literature that has been shortly reviewed. It is: at each abstraction level we must first know *what* information is needed, then decide *how* to extract it, and finally *analyse* the reliability and dependencies of the observations. This philosophy is reflected in the selections and motivations presented above.

The order of questions “what” and “how” means that we use psychoacoustic and physiological knowledge primarily for knowing what information should be observed in a music transcription system, not for knowing how to extract the information. In other words, the mechanism of the human ear must not determine the information of interest. Instead, we study *auditory cues* of a human listener in segregation and fusion of musical sounds, and search for analysis tools to extract that information.

How to know the relevant psychoacoustic information in music perception? Two excellent sources in the field of psychoacoustics are [Bregman90] and [Moore95]. What is the most suitable analysis method? The one that extracts the desired pieces of information as separately and precisely as possible. How to analyze the observations we obtain? Properties and dependencies of musical signals will be discussed at length in Chapter 6, other sources of error may be found in the analysis methods chosen.

In coming chapters, the different components of a transcription system are discussed in more detail.

4 Sound Onset Detection and Rhythm Tracking

“There is no doubt that music devoid of both harmony and melody can still contain considerable expression. Percussive music is a case in point, as anyone who has truly enjoyed traditional music from Africa, India, or Central or South America knows.”

-Jeffrey Bilmes [Bilmes93]

Expressive decisions in even performing a certain sequence of drum hits can carry a lot of content and meaning. This fact motivates the design of models that computers can use to produce *expressive* sounding rhythmic phrases instead of artificial quantized sequences [Bilmes93]. Further, interpreting the rhythmic role of each sound or finding the downbeat and upbeat automatically from a musical signal is anything but trivial.

But let us grossly drop out both understanding the expressive meaning and interpretation of the rhythmic role of each sound, trying first just to detect the onset times of the sounds. By an *onset time* we mean the time instant when a sound starts playing. As transcription of polyphonic music is concerned, we consider the detection of the onsets of the sounds by far easier than a subsequent analysis of each rhythmic segment. And truly: commercial devices have been released that synchronize to musical rhythm, but not even the first transcriber of polyphonic music.

Rhythmic meter is a framework in which musical events are located in time. It comprises beats, hierarchies and grouping. We did not have the possibility to study rhythm tracking with the same thoroughness and attempt as polyphonic transcription. This is why we practically stop at the level of sound onset detection and do not make a further analysis of the underlying rhythmic structure. Our original contribution to onset detection will also be quite limited.

4.1 Psychoacoustic bounds

A human ear is able to distinguish successive onsets of sounds from each other if they are separated by at least a 20 ms time interval, varying somewhat from an individual to another, and depending on the loudness of the sounds. If the interval is shorter, the former or louder onset *masks* the other one, and if there is a series of such successive onsets, they are perceived as a single rolling, low frequency sound. However, humans can detect even down to 5 ms *time deviations* in a rhythmic repetition of equidistant events [Bilmes93, Scheirer95].

We use the word *prominence* to refer to the strength at which an onset event stands out in a musical signal. The prominence of an onset is a result of several attributes: frequency of the attacking sound, its relative change in the amplitude, and the rapidity of the change.

4.2 Onset time detection scheme

A system capable of extracting rhythmic meter from musical signals is presented in [Scheirer96b]. On a corresponding web page, Scheirer provides a psychoacoustic demonstration on beat perception. The given set of audio signals shows that certain kinds of signal manipulations and simplifications can be performed without affecting the perceived rhythmic content of a musical signal. The demonstration is roughly as follows. First, a filter bank is

designed which divides a musical signal into six frequency bands, and the amplitude envelope of the signal at each frequency band is calculated. Then the corresponding frequency bands of a noise signal are controlled by the amplitude envelopes of the musical signal. For many kinds of filter banks, the resulting noise signal has a rhythmic percept which is significantly the same as that of the original music signal. Even with just four frequency bands, the pulse and meter characteristics of the original signal are easily recognizable. It should be emphasized that Scheirer's system is predominantly concerned with musical signals having a 'strong beat'. The above simplifications cannot be made for, e.g., classical music, which consists of non-percussive sounds only, such as a violin or an organ. A *percussive sound* is defined to be the sound of a musical instrument in which the sound is set on by striking, or by a short-lived contact of a driving force.

Since the only thing preserved in the above transformation is the amplitude envelopes of the filter bank outputs, it seems reasonable that only that much information is necessary to extract pulse and meter from a musical signal [Scheirer96b]. On the other hand, certain kinds of simplifications are *not* possible. For example, using only one frequency band, i.e., the amplitude envelope of the signal as a whole, will confuse the rhythmic percept. This observation led Scheirer to make a psychoacoustic hypothesis regarding rhythm perception: some sort of cross-band rhythmic integration, not simply summation across frequency bands, is performed by the auditory system. Based on the above considerations, Scheirer concludes that a *rhythmic processing algorithm should treat frequency bands separately, combining results in the end*, rather than attempting to perform beat-tracking on the sum of filter bank outputs.

We take this result as a foundation to our onset detection system, which therefore bears most resemblance to that of Scheirer. A couple of years earlier Bilmes proposed a system that was on a way to the same direction, but his system only used two bands, a high-pass and a low-pass filter, which was not as effective [Bilmes93]. Older rhythm tracking systems have typically used the amplitude envelope of a signal as a whole [Chafe85].

Distinct frequency bands and extracting their amplitude envelopes

As motivated above, the input signal is first divided into distinct frequency bands. We used seven one-octave bands and basically the same filter bank implementation as Scheirer. The filters are sixth order band-pass elliptic filters, the lowest being a low-pass filter and the highest a high-pass filter. Frequency band boundaries were set to 127, 254, 508, 1016, 2032 and 4064 Hz.

The output of each band is rectified (i.e., the absolute value of the signal is taken) and then decimated to ease the following computations. Amplitude envelopes are then calculated by convolving the signals with a 50 ms half-Hanning (raised cosine) window. This window performs much the same energy integration as the human auditory system, emphasizing the most recent inputs but masking rapid modulation [Scheirer96b, Todd92].

Calculating bandwise onset times

Until now, our system has been essentially analogous to Scheirer's one, but here we take our own direction. Both Scheirer and Bilmes calculate a *first order difference function* of the amplitude envelopes, taking the maximum rising slope of the amplitude envelope as the onset time of a sound [Bilmes93]. For us, it seems that a first order difference function is a good measure for the *prominence* of the onset, but its maximum values fail to precisely mark the

onset *times*. This is due to two reasons. First, especially low sounds may take some time to come to the point where their amplitude is maximally rising, and thus that point is crucially late from the physical onset of a sound. Second, the onset track of a sound is most often not monotonically increasing, and thus we would have several local maxima in the first order difference function near the physical onset (see plots with a dashed line in Figure 5).

We took an approach that effectively handles both of these problems. We begin by calculating a first order difference function, and take into account only the signal segments where the first order difference, i.e., the prominence of incoming activity, is above a certain threshold. Then we divide the first order difference function by the amplitude envelope function to get a first order *relative difference function*, i.e., the amount of change in relation to the absolute signal level. This is the same as differentiating the logarithm of the amplitude envelope.

We use the relative difference function to track onset times. This is psychoacoustically relevant, since the perceived increase in signal amplitude is in relation to its level, the same amount of increase being more prominent in a quiet signal. Moreover, the relative difference effectively handles the abovementioned problems by detecting the onset times of low sounds earlier and, more important, handling complicated onset tracks, since oscillations in the onset track of a sound do not matter in relative terms after its amplitude has started rising. To clarify this, we plotted the absolute and relative difference functions of the onset of a piano sound in Figure 5. Both of the benefits discussed can be seen clearly.

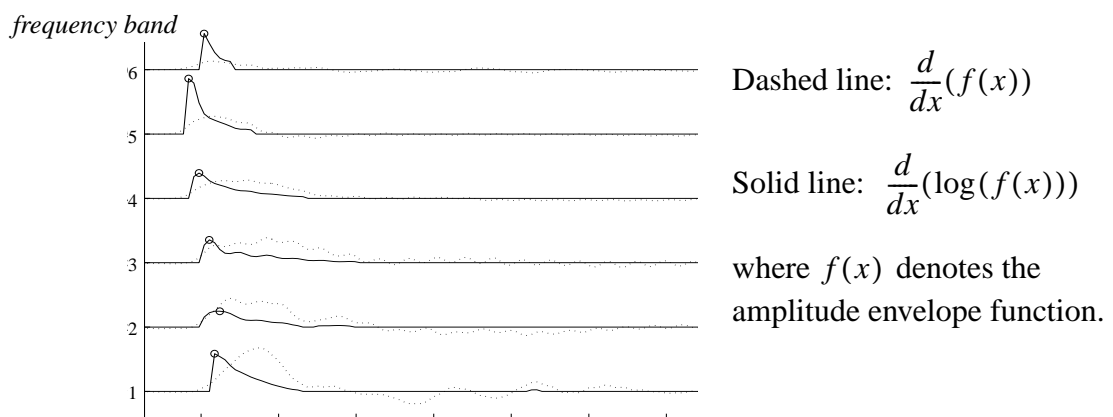


Figure 5. First order *absolute* (dashed) and *relative* (solid) difference functions of the amplitude envelopes of different frequency bands. Picked onset times are circled.

A set of potential onset *times* are collected from each band by performing a thresholded peak picking operation to the relative difference signals. The corresponding *prominences* of the potential onsets are found by taking the time of an onset, scanning forward to the next maximum in the *absolute* difference function, and taking the maximum value as the prominence.

Combining the results from different frequency bands

In the final phase we combine the results from different frequency bands to yield the onset times and prominences of the overall signal. First the potential onsets from different bands are all sorted in time order. Then each onset candidate is assigned a new prominence value, which is calculated by summing the prominences of onset candidates within a 50 ms time window around them. We drop out onset candidates whose prominence falls below a certain threshold

value. Then we also drop out candidates that are too close (< 50 ms) to a more prominent candidate. Among equally prominent but too close candidates, the middle one (median) is chosen and others are abandoned. Remaining onsets are accepted as true ones.

4.3 Procedure validation

The presented procedure was validated by testing its performance in finding sound onsets in polyphonic piano recordings and in rhythm music. Polyphonic piano recordings are separately taken into consideration, because our transcription system (see Chapter 7) uses piano music as simulation material, and will utilize the presented onset detection procedure for signal segmentation. The onset detection procedure was not tailored for each simulation case, but all presented results have been computed using the very same set of parameter values and thresholds.

Polyphonic piano recordings

Chosen piano compositions were played and recorded to evaluate onset detection efficiency. We used two microphones to pick the sound of an acoustic upright piano (Yamaha) and stored the signals on a digital audio tape using a 48kHz sampling rate. The microphone setup was according to a typical studio convention [Pawera81]. Note onsets were tracked in two excerpts of ten seconds, and the results are presented in Figure 6.

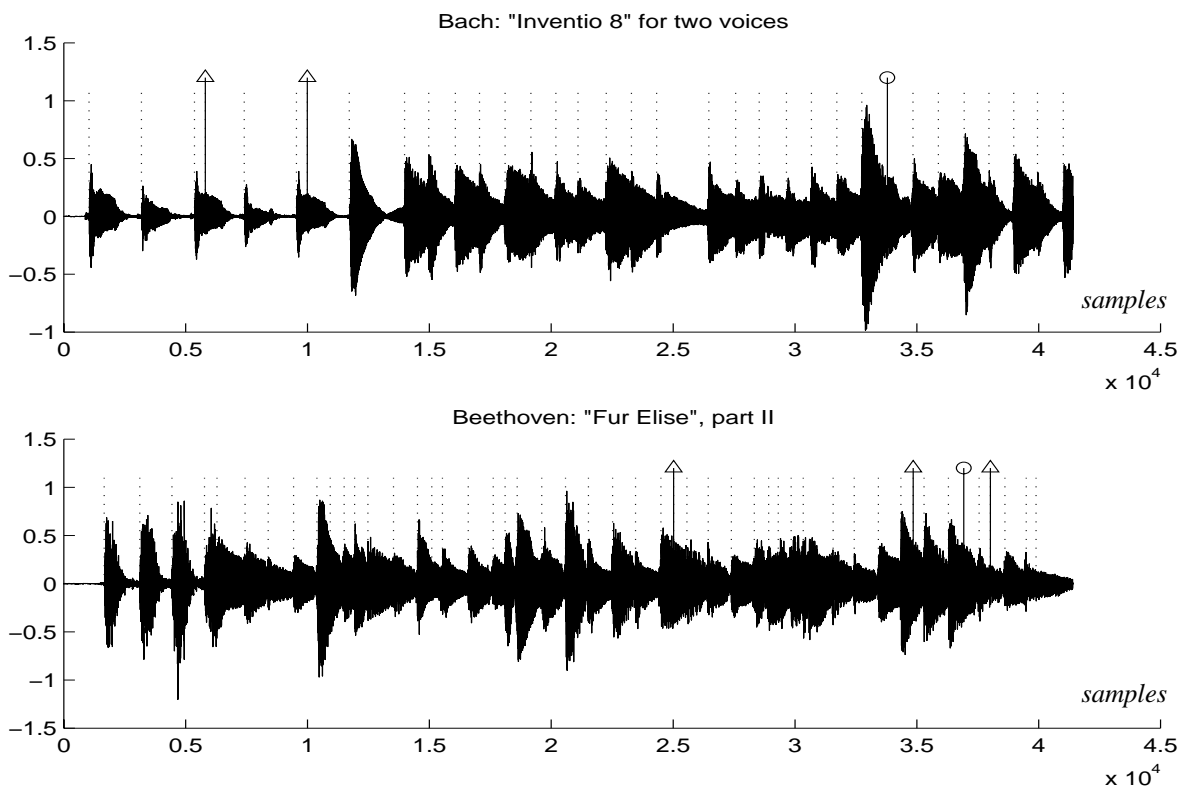


Figure 6. Tracked onsets in two piano compositions. Erroneous extra onsets are marked with arrows, missing onsets were added and are circled.

A recording comprising several instruments

Onset detection results are given for an example recording comprising several instruments and drums [Grusin91]. We consider one example sufficient, since reviewing the onset detection

results is a laborious task without the notes of the piece. The results are presented in Figure 7. This test case was successful, too. However, we noticed that more reliable detection of the onsets in musical signals that comprise different kinds of musical styles would call for rhythmic meter generation and rhythm analysis after onset detection (see Section 4.4). This would allow using prediction rules to recognise the weakest onsets and still not inserting erroneous extra onsets.

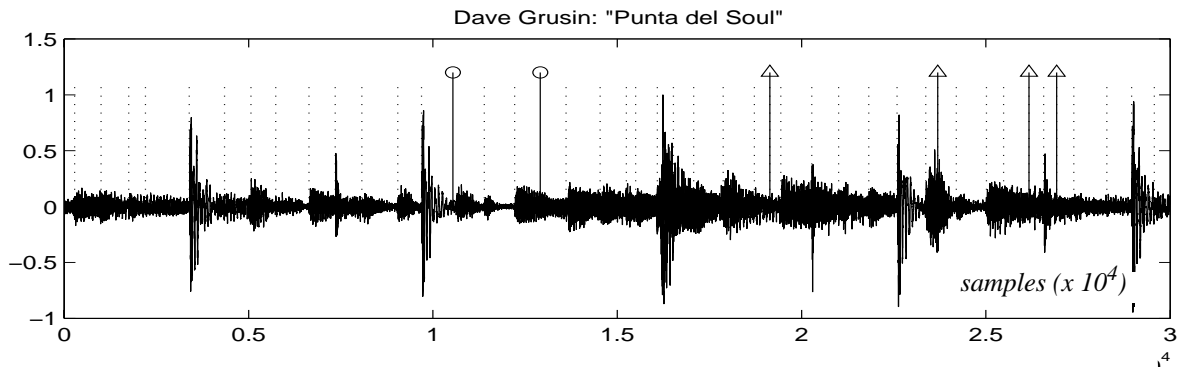


Figure 7. Tracked onsets in a recording comprising several instruments and drums. Erroneous extra onsets are marked with arrows, missing onsets were added and are circled.

4.4 Rhythmic structure

Several attempts have been made to understand the human rhythm perception and to automatically determine the rhythmic meter of a musical signal. Determining rhythmic meter means classifying the structure of a musical piece to *key signatures*, such as 3/4 or 4/4, and parsing musical events into units called *measures* that are separated from each other by vertical lines in musical notation [Brown93]. Still one step further is to understand the rhythmic role of each note. We summarize the different rhythm tracking proposals in Table 4, describe the objective of each, and indicate if they comprise onset detection in an acoustic signal or rhythmic structure determination, or both. Those that do not implement the former take their input in a symbolic form, for example in MIDI streams.

In his doctoral thesis, David Rosenthal describes a computer program which models the processes by which humans perceive rhythm in music [Rosenthal92]. His program reads a MIDI stream and gives a complete description of the rhythmic structure of the piece, its meter and the rhythmic role played by each note.

Jeffrey Bilmes aimed at designing algorithms and learning strategies that computers can use to produce expressive sounding rhythmic phrases, instead of artificial, exactly quantized sequences. He describes four elements with which he characterizes musical rhythm: metric structure, tempo variation, deviations, and ametric phrases. For a closer review on Bilmes's studies, see [Bilmes93].

Peter Desain and Henkjan Honing have published a lot about human rhythm perception and its computational modeling. To review their work see, for example, [Desain88,91,92,95]. Also several other approaches have been taken, the earliest being e.g. [Moorer75b] and [Chafe85]. Brown employed autocorrelation to determine the rhythmic meter in MIDI-streams [Brown93]. The abovementioned system of Scheirer's is unusual in the sense that it includes real-time acoustic pattern recognition, which allows the subsequent models to work with

Table 4: Rhythm tracking systems and the objective of each.

Reference	Onset detection	Rhythmic structure	Objective
Mont-Reynaud85		x	Finding rhythmic patterns in musical notations.
Rosenthal92		x	Givind a complete description of the rhythmic structure of a piece, and the rhythmic role of each note.
Brown93		x	Rhythmic meter determination.
Desain88,91		x	Rhythmic meter determination and quantization of the timings of each note.
Desain92		x	Theory on rhythm perception and definition of predictive expectancy of musical events.
Desain95		x	Computational modeling of the temporal structure in music cognition.
Bilmes93	x	x	Modeling performance nuances in rhythm, so that computers could produce <i>expressive</i> sounding rhythmic phrases.
Scheirer95	x		Extraction of the expressive performance information from an acoustic signal, when the musical notation of the piece is known in the process.
Scheirer96b	x	x	Determining the rhythmic meter of an acoustic musical signal.

acoustic data instead of symbolic representation.

We did not implement algorithms for recovering the rhythmic meter of musical signals, but stopped at the level of finding the onset times of sounds. This is done although rhythmic meter would allow utilizing predictive rules to detect weak onsets. Further, rhythmic structure plays an important role in top-down processing, where repetitions, redundancies and regularities of a musical signal are exploited in resolving its contents.

5 Tracking the Fundamental Frequency

This thesis is primarily concerned with the transcription of music which consists of *harmonic* sounds. Inharmonic sounds, such as drums, are detected and used in the rhythm tracking phase, but classifying them in the subsequent analysis is not discussed - we just refer to some transcription systems that perform that task (see Section 2.3 and Chapter 4).

From a music point of view, the most important attribute of harmonic sounds is their *fundamental frequency*, which allows arranging the sounds on a scale extending from low to high. Thus the most important requirement for a music transcription system is to detect the fundamental frequencies of the sounds, and write them down using the corresponding musical notation, notes.

There is a multitude of different methods for determining the fundamental frequency in an acoustic signal. In this chapter we make a short review of the different approaches by introducing a representative set of quite recent examples. Our intention is not to go to algorithmic details, and not even to provide a complete list of methods, but to give an idea of the potential approaches and principles that can be used in solving the problem. A more detailed comparative study can be found, for instance, in a relatively old but still relevant study of Rabiner et al. [Rabiner76].

Fundamental frequency tracking has almost exclusively been attempted for monophonic signals only. At the end of this chapter we will review some methods that have been used in analysing polyphonic signals.

5.1 Harmonic sounds

The physical appearance of a harmonic sound is a series of frequency partials called *harmonics*. They show up as peaks in the frequency spectrum (see Figure 8) at constant frequency intervals f_0 , with the lowest partial at frequency f_0 , which is therefore called the fundamental frequency of the sound. We use *pitch* as a synonym of the fundamental frequency, although there is a slight conceptual difference between them, pitch referring to the perceived fundamental frequency. In time domain, *periodicity* is a characteristic feature of harmonic sounds, $\frac{1}{f_0}$ being the fundamental period.

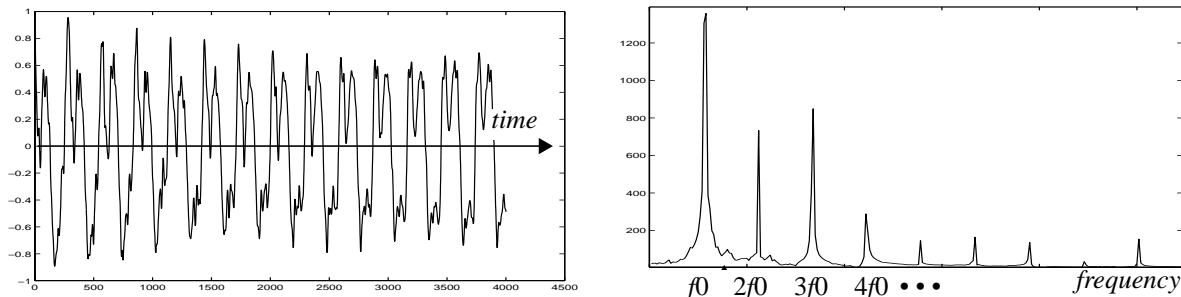


Figure 8. Harmonic sound in time and frequency domains.

A harmonic sound can be modelled as a sum of sinusoid partials

$$s(t) = \sum_{j=1}^N a_j(t) \cdot \cos[f_j(t) \cdot 2\pi t + \varphi_j] \quad (1)$$

where $a_j(t)$ are the amplitudes, $f_j(t)$ the frequencies, and φ_j phases of the partials. For strictly harmonic sounds, frequencies $f_j(t)$ are $j \cdot f_0(t)$, where $f_0(t)$ is the fundamental frequency. However, even the sounds of musical instruments may deviate a little from this rule during a short attack transient of the sound (approximately 50 ms in length), thus not being strictly harmonic.

5.2 Time domain methods

Autocorrelation-based

The autocorrelation function is particularly useful in identifying hidden periodicities in a signal. An estimate of the autocorrelation of an N -length sequence $x(k)$ is given by

$$r_{xx}(n) = \frac{1}{N} \cdot \sum_{k=0}^{N-n-1} x(k) \cdot x(k+n) \quad (2)$$

where n is a *lag*, or period length, and $x(n)$ is a time domain signal.

Many autocorrelation-based fundamental frequency trackers have been proposed. We take a quite recent example. In 1990, Brown published results of a study where the pitch of instrumental sounds was determined using autocorrelation [Brown91a]. She tried both conventional and ‘narrowed’ autocorrelation, which basically provides a better periodicity resolution in exchange to a weaker time resolution [Brown89].

The purpose of the study was to determine whether autocorrelation is well-adapted to fundamental frequency tracking of musical signals, and under which conditions ‘narrowed’ autocorrelation is advantageous. We skip the details concerning the calculation of the autocorrelation function and the selection of a winning periodicity in it [Brown91a]. What is most important here is that Brown suggests both conventional and narrowed autocorrelation to be good frequency trackers for musical sounds. However, autocorrelation is not very suitable for *polyphonic* musical signals because of the reasons that were discussed in Section 3.4. In addition, both Doval et al. and Rabiner remark that autocorrelation is not very suitable, when a large range of pitches and variety of spectra is encountered [Doval91, Rabiner76]. These attributes apply especially to music. For these reasons, we consider autocorrelation methods to be more adequate to speech than to musical signals.

Waveform-based

Some pitch detectors are based on time domain peak and valley measurements, or zero-crossing measurements. For example, Miller’s algorithm first detects *excursion cycles* in the waveform by observing intervals between major zero crossings [Miller75]. The remainder of the algorithm tries to identify the principal excursion cycles, i.e., those which correspond to true pitch periods. The algorithm was reported to work moderately well, although not as well as some other methods [Rabiner76]. Nowadays, the approach is practically of a historical value only.

5.3 Frequency domain methods

Harmonic matching pitch detection

As stated earlier, the appearance of a harmonic sound in the frequency domain is a series of equidistant energy maxima, at the positions of the harmonic partials. *Harmonic matching frequency domain pitch detection* means looking for a fundamental frequency, whose harmonics best explain the partials in a signal [Hess83].

A method proposed by Doval and Rodet belongs to this class [Doval91]. It is based on a probabilistic model for calculating the likelihood of an f_0 , when the frequency partials in a signal are known. The method is flexible enough to take into account possibly available prior knowledge of the sound in question. A mathematical expression for calculating the likelihood is quite complicated and can be found in the reference. The researchers paid particular attention to a large range of applicable f_0 s (50-4000 Hz), and to a very short response delay (20 ms). They tested the algorithm with musical and speech signals and proposed a real time implementation of it. *We made rather extensive simulations with Doval's algorithm, and it proved flexible and worked well.* The method employs a separate probabilistic treatment for the frequencies and amplitudes of each harmonic partial in the signal. This is important when the abovementioned objectives, large range of f_0 s and flexibility, are aimed at.

Doval and Rodet later published a higher level structure, which can be used for imposing constraints on the evolution of the signal, and for removing single gross errors in pitch detection by tracking the pitch contour through successive time frames [Doval93]. The higher level logic was based on hidden Markov models and the Viterbi algorithm. In the presented simulations, the gross error rate still decreased significantly when the higher level structure was employed.

Other frequency domain methods

There is a variety of methods that are more straightforward, or 'brute force', than the probabilistic approach of Doval et al. The structure of the spectrum of a harmonic sound prompts for revealing the fundamental frequency, the interval between the equidistant maxima, by applying autocorrelation to the magnitude of the spectrum. We take two examples. Lahat describes an algorithm, where the spectrum of the signal is flattened by a bank of bandpass filters, and the pitch is extracted from autocorrelation functions at the output of the filters [Lahat87]. This is somewhat more sophisticated than the straightforward autocorrelation of the spectrum, but utilizes the very same principle. Also Kunieda uses the autocorrelation function of the spectrum of a signal [Kunieda96]. Their method is called ACLOS (AutoCorrelation of LOg Spectrum), since they take the logarithm of a magnitude spectrum, flatten the result, and calculate its autocorrelation function to provide information of the pitch.

A *cepstral pitch detection* method has also been widely used, especially in extracting the pitch of speech signals [Noll64, Rabiner76]. A cepstral is calculated by first taking a short time Fourier transform of a signal segment, then calculating the logarithm of its magnitude, and finally inverse Fourier transforming the result. The peak cepstral value determines the pitch of the signal. The problem with cepstral pitch detection is that it cannot handle a very large range of pitches, which would be desirable in processing musical signals.

5.4 Utilizing prior knowledge of the sound

The earlier mentioned pitch detection method of Doval et al. is flexible enough to take into account prior knowledge of the sound in question [Doval91]. There are several other methods that rely even more heavily on prior knowledge of the sound source and the spectral distribution of the sounds. In the following, we take two examples.

In [Brown91b], Brown proposes a method that utilizes the fact that a constant Q spectral transform (see Section 3.5) gives a constant frequency domain pattern for harmonic sounds at different pitches. Thus she calculates the cross-correlation function between the constant Q spectrum and the ideal pattern of the sound in question, or just the pattern of one's at the positions of the harmonics. A peak in the cross-correlation function reveals the fundamental frequency. We thoroughly simulated the method in a transcription system that is reviewed on page 65. It proved to work fairly well for monophonic signals. Brown emphasizes that the method is able to deal with a variety of sounds and is consistent with the pattern matching theory of human pitch perception [Gerson78].

Spectra of musical sounds vary a lot and comprise ambiguous spectral energy distributions. The success of many proposed methods to cope with such sounds is often qualified by the need to carefully tailor parameters to yield desired results. Taylor and Greenough have developed a system which uses an adaptive resonance theory *neural network* called ARTMAP [Taylor93, Grossberg76]. In a preliminary phase, the network is *trained* by pitches from several instruments to achieve a tolerance to spectral variations. After training, the network can be used to *classify* new signals to their due pitches. Input to the network comes from logarithmically distributed frequency bins (12 per octave), output is a quantized pitch classification on a musical scale. The general idea of neural networks seems promising especially in tuning parameters of a system, but they do not remove the need for algorithm design, since the topology of a network sets limits to the model it can represent [Pao89].

5.5 Detection of multiple pitches

Determining the fundamental frequency of musical sounds has been relatively little explored in comparison to the massive efforts in estimating the pitch of speech signals for use in speech encoders and for communication purposes [Brown91a]. This is also the reason why detection of multiple pitches has been almost a neglected problem. Multipitch detection is particularly needed in the analysis of musical signals and in computational auditory scene analysis. Commercial interest of these areas is not as high as that of speech processing, especially when the difficulty of the problem is taken into account.

The need for a multipitch detector

It is generally admitted that the pitch detection methods discussed above are not appropriate as such to the detection of multiple pitches. This is especially true when musical signals are concerned, since very often the frequency relations of simultaneous sounds in music either make several sounds appear as a single coherent sound, or a non-existent 'ghost' sound strongly arise just because of the joint effect of the others. We reviewed this phenomenon in simulations with several pitch detection algorithms. Its cause was mentioned in Section 3.4, and will be more extensively discussed in Sections 6.2 and 6.3.

The problem cannot be solved by designing a sophisticated peak picking algorithm to follow a

single pitch detector. Consider, as an example, two sounds that are an octave apart in pitch from each other: the harmonic partials of the higher sound match perfectly the positions of even partials of the lower one, making it appear as a single sound, and turning the separation of the sounds into an even theoretically ambiguous problem.

In this section we represent methods that have been used to solve the problem. Awareness of it dates back to the very beginning of music transcription. In 1985, Chafe remarked: “Brute force detection schemes are less likely to succeed in analyzing polyphonic input - - - The balance between underdetection and overdetection becomes more difficult to achieve”. However, until very recently the problems of multipitch tracking have been controlled by heuristic ad hoc techniques, borrowing algorithmic principles from single pitch tracking. It is only in the past few years that seriously motivated methods have been proposed. They will now be discussed.

Tone model based processing

The abovementioned example of separating two sounds that are an octave apart in pitch from each other is theoretically ambiguous without prior knowledge of the sounds. It can be solved, however, if such information is collected to *tone models*, and utilized in signal analysis. Tone models are data structures that represent the spectral features of sounds in some format.

Two pitch tracking approaches that utilize knowledge of the involved sounds were introduced in Section 5.4. Kashino and Tanaka were the first to describe a complete tone model based music transcription system [Kashino93]. They also proposed an algorithm for *automatic tone modeling*, which means extraction of the tone models from the analyzed signal itself. The system suffered from some severe limitations, but was, on the contrary, more generic in regard to the types of instrumental sounds. In simulations, an advance registration of tone models had a significant effect on transcribing three-voice polyphonies, automatic tone modeling only worked in two-voice polyphonies.

Tone model based processing has also been used in the transcription of music which contains drum-like instruments only. Bilmes describes a quite sophisticated procedure for extracting the features of drum strokes, arranging them in a feature space, and using that information in a subsequent analysis of musical signals [Bilmes93]. Classification of the signal contents follows the lines of general pattern recognition.

Auditory model based

In Section 3.3 we discussed a variety of mid-level representations for a transcription system. There we also introduced multidimensional representations called correlograms and wefts. The motivation for these structures was that they not only explain a wide range of psychoacoustic phenomena in hearing, but also *try to organize sounds to their sources of production*, which in the case of music means distinguishing separate sounds.

Only one transcription system, that of Martin’s, uses a correlogram representation [Martin96a]. In the publication, Martin suggests that a correlogram facilitates the detection of two sounds that are an octave apart in pitch, even without introducing tone models. However, only little simulation evidence is presented to support that. We emphasize that when two sounds are in a straight harmonic relation, i.e., the fundamental frequency of the higher sound is a multiple of that of the lower sound, the two sounds cannot be unambiguously separated to two sounds by any bottom-up technique. That becomes possible, however, if there is a specific perceptual

cue, such as different onset times or specific amplitude / frequency modulations that associate harmonics to their sources. Correlogram representation does not pay any particular attention to those cues. Weft representation explicitly utilizes the common amplitude modulation cue, but by inspecting instrumental sounds we found out that this single cue can definitely not be assumed to be found in the sounds of all musical instruments [Ellis95].

As already suggested in Section 3.4, we conclude that although these psychoacoustically motivated models have proved successful in some computational auditory scene analysis tasks, they are not sufficient to solve the problem of music transcription. However, some benefits and conveniences may be attained by using these representations.

Explicit application of perceptual rules

A more conscious and controlled use of the known perceptual cues was first proposed by the forementioned research group of Osaka University [Kashino93]. In their publication, the group explicitly lists a collection of auditory cues that promote either fusion or segregation of simultaneous spectral features, thus telling whether certain frequency components belong to the same source or not. The cues had been collected from several sources, but only a limited subset of them were used in the simulated system.

Bregman lists the following cues in organizing simultaneous spectral features [Bregman90]:

1. Spectral proximity (closeness in time or frequency)
2. Harmonic concordance vs. harmonic mistuning
3. Synchronous changes of the components: *a)* common onset, *b)* common offset, *c)* common amplitude modulation, *d)* common frequency modulation, *e)* equidirectional movement in spectrum
4. Spatial proximity

None of the transcription systems that were introduced in Chapter 2 uses all of these cues. On the other hand, there is not even a theoretical consensus how these principles of perception *interact* and compete in human audition. It is also surprising that the fourth cue, *spatial proximity*, was not used in any of the transcription systems, although the human auditory system extensively utilizes it, music is mostly recorded stereo, and the stereo information is especially provided for separating the sound sources from each other in listening.

Knowing the significant pieces of information allows dividing the transcription problem into two steps: extraction of the relevant spectral features, and quantitative application of perceptual rules to assign the frequency component to their due sources of production. Osaka's group later also proposed an approach for the second step, i.e., *integration* of the evidence of the different auditory cues. They implemented a Bayesian probability network, which was employed to weight the different cues and to integrate the information from different knowledge sources [Kashino95].

We consider the 'auditory cue extraction' approach by far the most promising and generic in multipitch tracking. First, because utilization of universal auditory cues, in addition to tone models, makes the system more generic to different sound variations. Second, because explicit definition of the important cues separates information extraction from its interpretation, thus removing many underlying assumptions and allowing both quantitative and algorithmic integration of the cues.

5.6 Comment

Several approaches for detecting the fundamental frequency of monophonic signals were represented. Further, we discussed the need for specific tools to cope with polyphonic signals, in the analysis of which tone models and psychoacoustic cues were considered to be the most promising tools. We want to emphasize, however, that no general purpose multipitch tracking *algorithms* were found that would fit to the analysis of musical sounds.

The presented mechanisms do not remove the need to design multipitch tracking algorithms. Such are required to *resolve stable signals* that lack specific auditory cues, or to *observe the desired cues* reliably in polyphonic signals, or to make a sensible *association* between tone models and often quite inextricable spectra of musical signals. These matters are taken into consideration in the following chapter.

6 Number Theoretical Means of Resolving a Mixture of Harmonic Sounds

The next two chapters constitute the original part of this thesis, which means that the literature review type style will radically change. The reason for engaging in own research was the lack of publications in the area of multiple fundamental frequency tracking, or multipitch tracking. Indeed, we could not find any publications that would have focused on that matter, although in the monophonic case several algorithms have been proposed that are robust, commercially applicable and operate in real time.

Published efforts towards multipitch tracking have almost exclusively been made in the field of automatic transcription of polyphonic music. Until these days, however, transcription systems have been limited to two or three-voice polyphony only and to little generality of sounds. This is largely because no new mathematical foundation has been laid for polyphonic signal analysis, but the systems try to utilize the same principles in pitch tracking that are used in the monophonic case. We will discuss the spectral properties of a mixture of several harmonic sounds and demonstrate why single pitch tracking algorithms cannot be straightforwardly applied to polyphonic signals.

In this chapter we try to establish a number theoretical method to detect and observe harmonic sounds in polyphonic signals. This does not only concern multiple fundamental frequency tracking, but observing any of the features of harmonic sounds in polyphony. It is shown that a number theoretical approach is a vital additional level of analysis in any system where features of a harmonic sound are being observed in the presence of other harmonic sounds. The performance of the new methods will be evaluated when we apply them to the automatic transcription of piano music in Chapter 7.

6.1 Feature of a sound

We denote harmonic sounds with uppercase letters S and R . These are used consistently in such roles that sound S is being observed in the interference (presence) of a sound R , or R_i , if there are several interfering sounds. The attributes of harmonic sounds, both in time and in frequency domains, were illustrated in Section 5.1. From now on, we denote the harmonic partials, *harmonics* of a sound by h_j , where j goes from one up to the number of audible partials of the sound (see Figure 9). Braces are used to denote *sets*, thus $\{h_j\}$ being a set of harmonics.

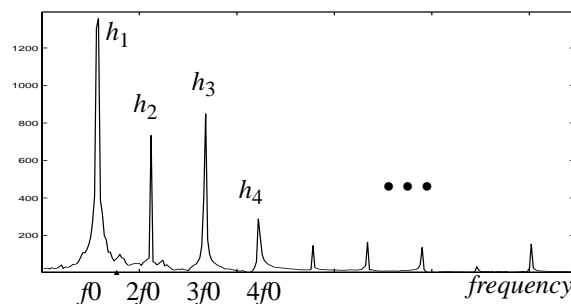


Figure 9. The way of denoting harmonic partials of a sound by h_j .

Further, we denote by $g(x)$ a *feature* of x , where x can be a sound S , or its single harmonic partial h_j . We will separate the different features by subscript characters, for example consistently denoting $g_F(x)$ the frequency of x , $g_A(x)$ the amplitude of x , and $g_L(x)$ the perceivable loudness of x . Explanation and definition of each feature will be given when they are taken into use.

We divide features $g(x)$ into two classes, depending on whether they can be associated with a single harmonic or not. Clearly, a single harmonic can have such features as frequency, amplitude, loudness and onset time, for example. This is the feature class I. To class II belong the features that can be applied to a sound, but are meaningless if x is a single harmonic h_j . As an example we consider *sound colour* $g_C(S)$, which results from the distribution of the energy of a sound to its harmonics. Now, a single harmonic does not have such a feature as colour, and also cannot represent this feature of the sound alone. Thus the features of a sound that belong to class I can be approximated by independent representative harmonics, but class II requires observation of the whole harmonic series as a single entity.

Because the very substance of a harmonic sound is its series of equidistant sinusoid partials, any *observation of a harmonic sound must be based on its harmonic partials*, since strictly speaking the sound has no other physical appearance than these partials. Calculation of any of the features of the sound must be based on the features of its harmonics, no matter if this is done in time or frequency domain.

6.2 The basic problem in resolving a mixture of harmonic sounds

There is a multitude of good methods for measuring the frequencies, amplitude envelopes and phases of sinusoid partials in a signal [McAulay86, Depalle93, Serra97, Ding97, Qian97]. The problem in resolving a mixture of harmonic sounds is that the harmonics of a sound extend to a wide frequency band, and most often the frequency bands of different sounds overlap, and several harmonics of different sounds overlap. This causes two severe problems.

1. Because the harmonic series of different sounds typically extend to common frequency bands, it is most difficult to assign the harmonics to their true fundamental frequencies.
2. If two sinusoid partials overlap, i.e., have a common frequency, the amplitude envelopes and phases of the overlapping sinusoids can no longer be deduced from their sum.

These are the fundamental reasons why polyphonic signal contents cannot be resolved by straightforwardly applying the algorithms designed for monophonic signals.

Proposition 1. If any harmonic h_j^S of a sound S is overlapped by any harmonic h_i^R of an interfering sound R , then the fundamental frequency of the sound R must be $f0_R = \frac{m}{n} \cdot f0_S$, where m and n are integer numbers greater than or equal to 1.

Proof. The frequency of a harmonic h_j^S of the sound S is

$$g_F(h_j^S) = j \cdot f0_S \quad (3)$$

and the frequency of a harmonic h_i^R of the sound R is

$$g_F(h_i^R) = i \cdot f0_R$$

The condition of a harmonic h_j^S of a sound S to be overlapped by a harmonic h_i^R of an interfering sound R can be expressed as

$$i \cdot f0_R = j \cdot f0_S \Leftrightarrow \left(f0_R = \frac{j}{i} \cdot f0_S \right), \quad (4)$$

when the common factors of j and i are reduced, this can be expressed as

$$f0_R = \frac{m}{n} \cdot f0_S, \quad (5)$$

where $(m, n) \geq 1$ and can be calculated from the numbers of the harmonics, i and j . \square

Comment. It may look as being a special case that the fundamental frequencies of two sounds would have exactly this relation, but it is definitely not. This is due to the physics of sound production and the rules governing fundamental frequencies of the notes in Western music, where the fundamental frequencies of the notes most often have the abovementioned relation, m and n being small integers.

Proposition 2. If the fundamental frequencies of two harmonic sounds S and R are $f0_S$ and $f0_R = \frac{m}{n} \cdot f0_S$, respectively, then every n^{th} harmonic h_{nk} , $k = 1, 2, 3, 4, \dots$ of the sound R overlaps a corresponding m^{th} harmonic h_{mk} , $k = 1, 2, 3, 4, \dots$ of the sound S .

Proof. Substituting (5) to (4) we can rewrite the condition of a harmonic h_j of a sound S to be overlapped by a harmonic h_i of an interfering sound R as

$$(i \cdot f0_R = j \cdot f0_S) \Leftrightarrow \left(i \cdot \frac{m}{n} \cdot f0_S = j \cdot f0_S \right) \Leftrightarrow (i \cdot m = j \cdot n),$$

which is true for each pair $i=nk$ and $j=mk$, where $k=\{1, 2, 3, 4, \dots\}$. \square

Corollary. If the fundamental frequency of a sound R is $f0_R = \frac{1}{n} \cdot f0_S$, it will *overlap all the harmonics of S* at their common frequency bands. Otherwise R can overlap maximally every second harmonic of S . If all harmonics of S are overlapped by an interfering sound R , detecting the two sounds is difficult and even theoretically ambiguous: the harmonics of S might have resulted from R alone, and the sound S is easily undetected. On the other hand, if the loudness of the sound S is deliberately determined regardless of the sound R , S will ‘steal’ the energy of every n^{th} harmonic of R to itself.

6.3 Certain principles in Western music

“Music transforms arithmetic relations into immediate sensations.”

-Schopenhauer, Metaphysics of Music

An important principle governing music is paying attention to the frequency relations, *intervals*, of simultaneously played notes. Here we use the word note to refer to a sound that is produced by a musical instrument. Paying attention to time intervals in rhythm and to frequency intervals of concurrent sounds has a certain goal: to unify the sounds to form a coherent structure that is able to express more than any of the sounds alone.

Two notes are in a *harmonic* relation to each other if their fundamental frequencies satisfy

$$f0_2 = \frac{m}{n} \cdot f0_1, \quad (6)$$

where m and n are small integers. The smaller the values of m and n are, the closer is the harmonic relation of the two sounds and the more perfectly they play together. For example the fundamental frequencies $\frac{4}{4}f$, $\frac{5}{4}f$ and $\frac{6}{4}f$ constitute a basic *major* chord, and the fundamental frequencies $\frac{4}{6}f$, $\frac{4}{5}f$ and $\frac{4}{4}f$ constitute a *minor* chord. To read more about the mathematical viewpoint to music, see [JIN98] and [Canright87].

The notes in music are arranged to a logarithmic scale, where the fundamental frequency of a note k is

$$f0_k = 440 \cdot 2^{\frac{k}{12}} \text{ Hz} \quad (7)$$

and the notes in a standard piano keyboard range from $k = -48$ up to $k = 39$. Although the scale is logarithmic, it can surprisingly well produce different *harmonic intervals* that can be derived from Equation (6) by substituting small integers to m and n . To highlight this we calculate the frequency relations of two notes inside a single octave of notes in Table 5.

Table 5: Comparison of realizable frequency intervals and ideal harmonic relations.

Note 1	Note 2	$f0(N_2) : f0(N_1)$	m	n	$m : n$	Deviation from harmonic relation
$C\#$	C	1.0595	16	15	1.0667	-0.68 %
D	C	1.1225	9(8)	8(7)	1.125(1.143)	-0.23 % (+1.8 %)
$D\#$	C	1.1892	6	5	1.2	-0.91 %
E	C	1.2599	5	4	1.25	+0.79 %
F	C	1.3348	4	3	1.3333	+0.11 %
$F\#$	C	1.4142	7	5	1.4	+1.0 % : <i>a lot</i>
G	C	1.4983	3	2	1.5	-0.11 %
$G\#$	C	1.5874	8	5	1.6	-0.79 %
A	C	1.6818	5	3	1.6667	+0.91 %
$A\#$	C	1.7818	16(7)	9(4)	1.778(1.75)	+0.23 % (-1.8 %)
H	C	1.8877	15	8	1.875	+0.68 %

The realizable musical intervals deviate a little from the ideal harmonic relations, but the amount of error is so little that practically it does not disturb the human ear. Relations $f0(F\#) : f0(C)$ and $f0(A\#) : f0(C)$ make an exception, and indeed, the term *blue note* in music refers to situations, where a note is played in a pure natural frequency relation instead of the closest ‘legal’ note. This is most often done for the seventh of a chord ($A\#$ for C), which would otherwise deviate 1.8 % from its ideal.

For a practical frequency resolution the overlapping of the harmonics of the two sounds is the same as if the harmonic relation were perfect. Therefore every third harmonic of a note F will overlap every fourth harmonic of a simultaneously played note C , for example. We take an example to clarify how the harmonics of simultaneously played notes overlap each other. The analyzed basic major chord is of fundamental importance in Western music.

Example 1. Analysis of the harmonic mixture of a C major chord, which is built from the notes C , E and G . The results apply for any major chord. The fundamental frequencies of the notes are f , $\frac{5}{4}f$ and $\frac{3}{2}f$, where constant f is determined by the fundamental frequency of the chord root, but can be ignored, when frequency *relations* of the notes are considered.

Notes E and G are in $\frac{5}{4}$ and $\frac{3}{2}$ relations to C , respectively, and based on Proposition 2 they overlap every fifth and every third harmonic of C , respectively. This causes 47% of the harmonics of the note C to be overlapped and - from the viewpoint of observing the

sound C - confused. Similarly, notes C and G are in $\frac{4}{5}$ and $\frac{6}{5}$ relations to E , and overlap every fourth and every sixth harmonic of E , causing 33% of the harmonics of E to be overlapped. Further, notes C and E are in $\frac{2}{3}$ and $\frac{5}{6}$ relations to G , and together overlap 60% of the harmonics of the note G .

Consider now determining the existence of any of these notes, or observing their features. Sixty percent of the harmonics of the note G would be found from the spectrum even in the absence of the note G ! On the other hand, the problem is how to extract the features (e.g. the loudnesses) of the notes, when approximately half of their harmonics are strengthened by the harmonics of the other two notes.

Analysis of several other typical note combinations is presented in Table 6. Again, the combinations are derived from note C , but apply to any chord of the same class. This means that the results are the same for C^0add6 and F^0add6 , for example.

Table 6: Analysis of overlapping harmonics in typical note combinations

Chord	Notes in the chords				Note frequencies in $m:n$ form, respectively					Percentage of overlapped harmonics (%), maximum bolded					Average	
C_{major}	c	e	g		1	5:4	3:2			47	33	60			47	
C_{minor}	c	d#	g		1	6:5	3:2			33	20	50			33	
C^0add6	c	d#	f#	a	1	6:5	7:5	5:3		43	31	33	33		35	
C^+	c	e	g#	c2	1	5:4	8:5	2		60	25	30	100		54	
C^9	c	d	e	g	a#	1	9:8	5:4	3:2	16:9	50	23	39	65	11	38
$c+C_{major}$	c1	c3	e3	g3		1	4	5	6		47	100	100	100		87
$inharm.$	c	c#	d	d#	e	1	16:15	9:8	6:5	5:4	41	11	20	25	33	26

These examples illustrate why the algorithms designed for the detection and observation of a single harmonic sound do not apply to resolving polyphonic musical contents. Further, it can be seen that an additional peak picking or logical level after such an algorithm is not a solution, since observations get confused already at the lowest level. Instead, we need to rethink the very kernel, how to collect the information of a sound from its harmonics.

6.4 Prime number harmonics: an important result

Prime number harmonics $\{h_1, h_2, h_3, h_5, h_7, \dots\}$ of a sound share a desired common property that is derived from the very definition of the prime numbers: they are divisible only by one and themselves. This has an important consequence, which will give a steadfast starting point in organizing the jungle of sinusoids to their due fundamental frequencies.

Proposition 3. Any harmonic sound R can overlap only one prime number harmonic of a sound S , provided that the fundamental frequency of R is not $f0_R = \frac{1}{n} \cdot f0_S$, where $n=1,2,3,4,\dots$. In the case where the fundamental frequencies are in special relation mentioned, the sound R overlaps *all* the harmonics of S .

Proof. This can be proved by assuming that two prime number harmonics of S are overlapped by the harmonics of R and showing that in this case $f0_R = \frac{1}{n} \cdot f0_S$, where n is an integer

greater than or equal to 1, and the sound R overlaps all the harmonics of the sound S .

Let $f0_S$ and $f0_R$ be the fundamental frequencies of the sounds S and R , respectively. We denote an arbitrary prime number by p_j . The condition of two prime number harmonics of S being overlapped by any harmonics h_i of R can be expressed as

$$\begin{cases} i_1 \cdot f0_R = p_1 \cdot f0_S \\ i_2 \cdot f0_R = p_2 \cdot f0_S \end{cases}, \quad (8)$$

where p_2 can be solved as

$$p_2 = \frac{p_1 \cdot i_2}{i_1}.$$

In order for p_2 to be a prime number and not equal to p_1 , i_1 must satisfy

$$i_1 = n \cdot p_1, \quad (9)$$

where n is an integer implying

$$i_2 = n \cdot p_2.$$

Substituting (9) to (8) we get

$$f0_R = \frac{p_1 \cdot f0_S}{i_1} = \frac{p_1 \cdot f0_S}{n \cdot p_1} = \frac{f0_S}{n}, \quad (10)$$

where $n \geq 1$. \square

If Equation 10 holds, all harmonics of $f0_S$ are overlapped by the harmonics of $f0_R$, since

$$\begin{aligned} i \cdot f0_R &= j \cdot f0_S \\ f0_R &= \frac{f0_S}{n} \Rightarrow i \cdot \frac{f0_S}{n} = j \cdot f0_S \Leftrightarrow i = n \cdot j, \end{aligned}$$

which means that all harmonics of S are overlapped by every n^{th} harmonic of R .

6.5 Dealing with *outlier* values

Let us denote the set of prime harmonics by $\{h_p \mid p \text{ is prime}\}$, and the set of the features of the prime harmonics by $\{g(h_p) \mid p \text{ is prime}\}$, where the type of the feature is not yet fixed. Prime number harmonics of a sound S can be considered as *independent pieces of evidence for the existence of the sound S, or for any of its features* that can be deduced from its harmonics. This is because each of the other harmonic sounds in the signal can overlap no more than one prime number harmonic of S , provided that they do not have a fundamental frequency in the above-mentioned harmonic relation $f0_R = \frac{1}{n} \cdot f0_S$ under S .

In the set of representative features $\{g(h_p) \mid p \text{ is prime}\}$ there are two kinds of *outliers*, i.e., irrelevant values in respect of the true feature $g(S)$ of the sound: some prime harmonics have been disturbed by interfering sounds, while some others are irrelevant because of the properties of the sound S itself, for example some prime harmonics may be totally lacking from S . Those values that are *not* outliers vary somewhat in value, because of irregularities and noise, but outlier values are characterized by the fact that they are single, clearly deviated values, and invalid to represent the true feature of S .

Example 2. Figure 10 illustrates a situation where the *onset time* $g_T(S)$ of a note d (293 Hz) is being determined, when an interfering note a ($\frac{3}{2} \cdot 293$ Hz) has set on a little earlier. As stated in Proposition 2, every second harmonic of note a overlaps every third

harmonic of note d , but as stated in Proposition 3, only one prime harmonic is overlapped, the third one. The sinusoids at the positions of the first ten harmonics of note d have been indicated by lines, and the dots indicate their onset times. Overlapped harmonics 3 (prime), 6 and 9 are outliers, since onsets of these sinusoids are due to the interfering note a , and propose all too early onset times. On the other hand, harmonics 8 and 10 were not detected, and represent all too late onset times, thus being outliers, too.

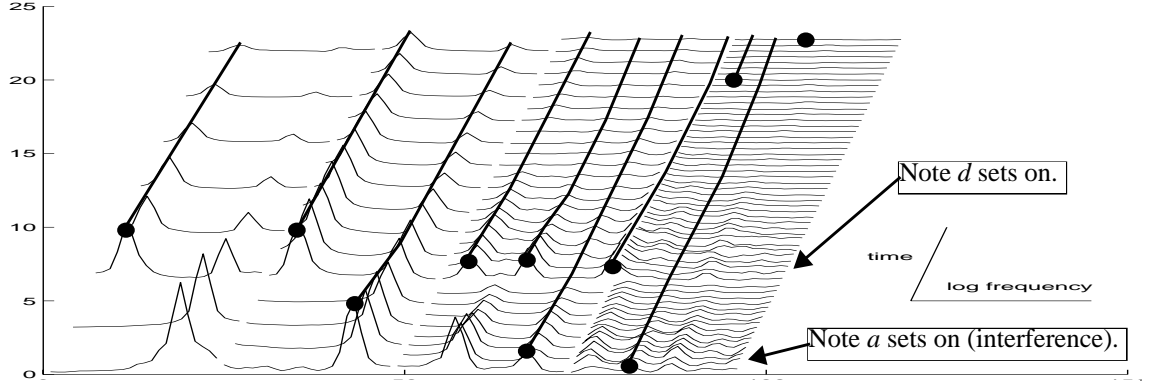


Figure 10. Determining the sound onset time in interference of an earlier started sound.

An analogous situation is met when e.g. the *loudness* $g_L(S)$ of a sound is being observed: some single prime harmonics have been overlapped by interfering sounds, and are outliers. In this case missing harmonics do not introduce true outliers - their loudness is just weaker than expected (zero), and are compensated by other harmonics that are louder than expected.

We are now faced with data in which several independent pieces of evidence represent the same quantity. Some of them are corrupted, outlier values, but a majority of the representatives, called *samples*, are reliable, it being highly improbable that a majority of the prime number harmonics would be either missing or each corrupted by an independent interfering sound. *This is the motivation for a design of a filter which would pick the estimated feature $\hat{g}(S)$ from the set of independent representatives $\{g(h_p) \mid p \text{ is prime}\}$, and drop out the irrelevant values.*

Consider taking the mean among the representatives, $mean\{g(h_p) \mid p \text{ is prime}\}$. This is not satisfactory, since even one outlier in a set of, for example, seven prime harmonics will cause an unacceptable deviation to the mean, since outlier values may be grossly erroneous. Instead, the class of *median and order statistic filters* is prompted by the fact that they are particularly effective in dealing with the kind of data that we characterized above. These filters depend on *sorting* the set of representatives. In our previous Example 2, sorting the onset times of the prime harmonics would result as depicted in Figure 11.

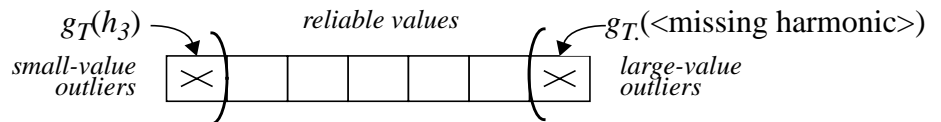


Figure 11. Outlier values map to the both ends of a sorted set of representatives.

It can be seen that under- or overestimated outlier values map to the both ends of the sorted set, and in between, the reliable representatives are sorted from the smallest up to the largest. Because we can assume the majority of the values to be reliable, we can safely pick, for example, the middle (median) value of the sorted set. Thus we yield the desired feature of the sound

by

$$\hat{g}(S) = \text{median}\{g(h_p) \mid p \text{ is prime}\}. \quad (11)$$

More knowledge of the feature in question, and of the nature of the outliers - for example knowing that only large-value outliers exist in data, would perhaps suggest picking some other than the middle value. This and some other questions will be discussed more extensively in the following section, where order statistic filters are introduced. For now, it is most important to notice that sorting the set of representatives is a good way to separate outlier values.

6.6 Generalization of the result and overcoming its defects

The fundamental building block for our music transcription algorithm has been stated. And truly: in practical simulations picking the median of the features of prime number harmonics to represent the whole sound proved to be robust in rich polyphonic contents, when automatic transcription of polyphonic musical contents was attempted.

There are still inevitable defects to overcome, however. *Too few harmonics* are taken into account in calculations, if only the prime number harmonics are used to determine a feature of a whole sound. This degrades the usability of the algorithm and makes it sensitive to the tonal content of a sound, where even all prime number harmonics could be missing. A small set of representative harmonics also produces large deviations in the result, although its value would be reliable in the sense that gross errors are avoided even in interference.

The other notification to be made is that all the harmonics in a set of harmonics are not equally trustworthy in value, but we need to *emphasize the lowest harmonics*. This is due to three reasons. First, the amplitude of the higher harmonics is generally smaller and thus more sensitive to interference and noise. Second, if the sound source is not known, we cannot even know the precise number of accountable harmonics. The third reason is met when matching a pattern of a tone model to a signal: our tone model is typically rough and sound variations depending e.g. on the instrument and playing velocity especially affect the value of the higher frequency harmonics which no longer resemble the tone model used.

We proceed towards a model where these shortcomings are removed but the advantages of the set of prime number harmonics are preserved. We denote by ν a feature extraction filter to perform the operation

$$\hat{g}(S) = \nu\{g(h_j)\} \quad , \quad (12)$$

where $\{g(h_j)\}$ is the set of features of all the harmonics of the sound S , each being a representative of a feature of the whole S . $\hat{g}(S)$ is the estimated feature of the whole sound S . This means that ν picks the estimated feature of a sound from the set of features of its harmonics. The requirements of the filter ν can now be exactly expressed as given in Table 7.

The use of the class of median and order statistic filters was motivated in Section 6.4. In his doctoral thesis, Kuosmanen has developed algorithms for the statistical analysis of stack filters, comprising the abovementioned filter classes plus an important portion of other non-linear filters [Kuosmanen94, Astola97]. We implemented Kuosmanen's algorithms, and will use them in the continuation. First we need to define two concepts. The j^{th} *sample selection probability* of a filter ν means the probability that the sample h_j in a set $\{h_j\}$ is selected to be the output of $\nu\{h_j\}$. We denote sample selection probability by $P_s(j)$. The i^{th} *rank selection probability* of a

Table 7: Requirements of the feature extraction filter.

1. A single interfering sound may cause an overlapped (disturbed) harmonic to exist in output with only up to a *limited* probability. More generally this reads: given a number of N interfering sounds may together cause a disturbed outlier harmonic to be chosen to the output of ν with only up to a given limit probability λ (lambda).
2. The filter must utilize all the harmonics of the observed sound as equally as possible to make it applicable and robust to different kinds of sounds. Picking *only* the prime number harmonics is not very satisfying from this point of view.
3. The filter should have *as a high breakdown point as possible*, which means the number of outlier values that are needed to break down the statistics of the filter, i.e., the output of the filter is also an outlier [Kuosmanen94].

filter ν means the probability that the i^{th} smallest value in an ordered set $\{h_j\}$ is selected to be the output of $\nu\{h_j\}$. We denote the rank selection probability by $P_r(i)$.

Several design efforts were made with so-called *multistage median* filters, which will not be discussed here, since a thorough statistical analysis of different kinds of stack filters showed that *weighted order statistic filters* (WOS) have the best statistical properties for our use [Astola97]. The output of a WOS filter is calculated as depicted in Figure 12: representatives (samples) are first sorted according to their value, then each sample is repeated according to its *weight* that is given in the filter's parameters, and finally the T^{th} smallest value is chosen from the resulting sorted multiset. T is the threshold, and also a parameter of the filter [Kuosmanen94]. From now on we will use WOS filters only, since their weights allow convenient tailoring of the sample selection probabilities, i.e., the probabilities of the harmonics to be picked to represent the whole sound. The threshold T , in turn, can be used set to bias rank selection probabilities around the median or some other value.

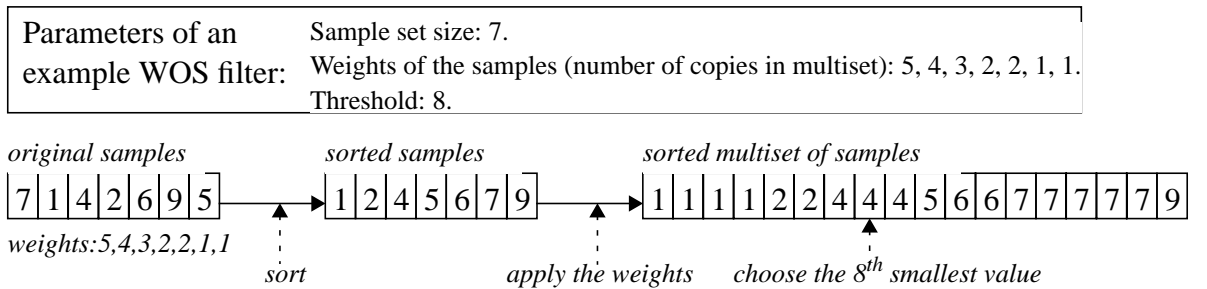


Figure 12. Parameters and calculations of an example WOS filter. $\nu\{7,1,4,2,6,9,5\} = 4$.

We need one more definition still. A set E_m is defined to be

$$E_m = \{h_{m \cdot j}\}, \tag{13}$$

where j is an integer greater than or equal to one. In other words, the subset E_m contains every m^{th} harmonic, starting from harmonic m .

On page 32, we proved that if an interfering sound R overlaps a harmonic of an observing sound, it overlaps every m^{th} harmonic of it, i.e., exactly the subset E_m . Therefore we can now rewrite the first requirement of Table 7 as finding the sample selection probabilities $P_s(j)$ for

the filter v so that the selection probabilities of the N largest subsets E_m together sum up to the given limit probability λ . N largest subsets, which are not subsets of each other are the prime sets $\{E_m \mid m=2,3,5,7,\dots\}$. E_1 is excluded, because the case of all harmonics being disturbed will be discussed separately in Section 6.9. If N is set to 1 this can be expressed as finding $P_s(j)$ in a minimizing problem

$$\min \left\{ \max_{m \geq 2} \left\{ \sum_{i=1}^{\infty} P_s(m \cdot i) \right\} \right\} . \quad (14)$$

The second requirement can be expressed in these same terms as trying to distribute the unity sample selection probability as equally as possible to all harmonics. The third requirement calls for implementing as tight a rank selection probability as possible, i.e., trying to concentrate the rank selection vector P_r around the T^{th} smallest value. Actually it can be shown that satisfying the second requirement will result in satisfying the third as well. See Figure 15 on page 45 to have an idea of what sample and rank selection probability vectors could look like.

We will first develop an algorithm for finding the desired sample selection probabilities $P_s(j)$, and later design a WOS filter to implement that $P_s(j)$.

Finding the desired sample selection probability vector

We have earlier stated that an interfering sound R overlaps harmonics of a sound S if and only if ${}^c0_R = \frac{m}{n} \cdot f0_s$, and in this case it overlaps the subset E_m among the harmonics. Here we assume all fundamental frequencies of interfering sounds to be equally probable. Further statistical analysis of music might prove that this assumption is not true, but we prefer not to make music style specific assumptions, and on the other hand do not have the possibility to make a more precise analysis of the typical distribution of R_i 's.

Based on the assumption, all m and n values binding the fundamental frequency of R are equally probable (up to the total number of harmonics, since otherwise the probability of a certain m would approach zero), from where it follows that R is equally probable to choose to overlap any subset E_m . However, the *relative trustworthiness* is not the same for all the single harmonics h_j , but equals the probability that none of the sets E_m that h_j belongs to is overlapped. This can be calculated as $\tau^{D(j)}$, where τ represents the overall probability of an interfering sound to overlap some subset E_m and $D(j)$ is the number of subsets E_m that harmonic h_j belongs to. It can be easily proved that $D(j)$ is the number of integers that divide j , $D(1)=1$. An integer a is defined to *divide* another integer b , if and only if $b = da$ holds for some integer d [Koblitz87].

Sample selection probabilities of the harmonics should be according to their probability of being trustworthy. We can therefore write $P_s(j)$ in the form

$$P_s(j) = \tau^{D(j)} , \quad (15)$$

where $j \geq 1$, and $D(j)$ is as defined above.

We denote the total number of harmonics of the observed sound by J . The overall selection probability of the harmonics is

$$\sum_{j=1}^J \tau^{D(j)} \quad (16)$$

and should equal to unity.

Set I is defined to contain the numbers j of the harmonics h_j that belong to some of the N largest subsets $\{E_m \mid m=2,3,5,7\dots\}$. If $N=1$, set I simply contains even numbers up to J . The sum over the selection probabilities of the harmonics in these N largest subsets is

$$\sum_{j \in I} \tau^{D(j)}. \quad (17)$$

Finally, we can rewrite the first requirement of the feature extraction filter ν (see Table 7 on page 38) as

$$\sum_{j \in I} \tau^{D(j)} = \lambda \cdot \sum_{j=1}^J \tau^{D(j)}, \quad (18)$$

from which τ can be solved. Selection probabilities $P_s(j)$ can now be uniquely solved by substituting τ to Equation 15.

The special properties of the resulting $P_s(j)$, and the role of the selected λ value will be discussed after representing the above reasoning in an algorithm form.

Algorithm 1. Calculation of the sample selection probabilities $P_s(j)$ of a feature extraction filter ν , so that the selection probability of a harmonic is according to its relative probability of being trustworthy, $\tau^{D(j)}$, and so that the sum over the selection probabilities of the harmonics in N largest subsets $\{E_m \mid m=2,3,5,7\dots\}$ yields the given limit probability λ . The sum over whole $P_s(j)$ is unity.

Step 1. Calculate $D(j)$, where $j=1,2,\dots,J$, so that $D(j)$ is the number of positive integers that divide j . This is the number of subsets E_m that a harmonic h_j belongs to.

Step 2. Form set I , which contains the numbers j of the harmonics h_j that belong to some of the N largest subsets $\{E_m \mid m=2,3,5,7\dots\}$. If $N=1$, set I simply contains even numbers up to J .

Step 3. Set $emax = \max\{D(j)\}$. Calculate vectors $A_1(e)$ and $A_2(e)$, where $e=1,2,\dots,emax$, so that

$$A_1(e) = \text{number of items in } D(j) \text{ that satisfy } D(j) = e$$

$$A_2(e) = \text{number of items in } \{D(j) \mid j \in I\} \text{ that satisfy } D(j) = e.$$

Step 4. Form polynomial $f(\tau) = a_1 \cdot \tau + a_2 \cdot \tau^2 + \dots + a_{emax} \cdot \tau^{emax}$, where the coefficients a_e are $a_e = A_2(e) - (\lambda \cdot A_1(e))$.

Step 5. Find all roots of this polynomial by solving $f(\tau)=0$. Although this cannot be done analytically for higher than fourth order polynomials, efficient numerical methods exist. In practice, this reduces to finding eigenvalues of the associated companion matrix of A [Horn85].

Step 6. If the problem is solvable for given N , λ and J , there is only one root that is real and between 0 and 1. This root is the value τ that was discussed earlier.

Step 7. Calculate an initial sample selection probability vector of the WOS filter as

$$\hat{P}_s(j) = (\tau)^{D(j)}.$$

Step 8. Divide each value $\hat{P}_s(j)$ by $\sum \hat{P}_s(j)$ to scale its sum to unity. This is $P_s(j)$.

We arrive at sample selection probability values $P_s(j)$, where

$$\sum_{j \in I} P_s(j) = \lambda,$$

and set I is as defined earlier. This means that N interfering sounds may together contribute only up to λ probability that an overlapped harmonic exists in the output. Another very important property of this algorithm is that we can flexibly make a tradeoff between the requirements 1 and 2 in Table 7. Concretely, this means that the less we put emphasis on the robustness of the filter ν in the presence of interfering sounds, the more equally the filter utilizes all the harmonics of the observed sound, and vice versa. Some different parameter selections are illustrated in Figure 13.

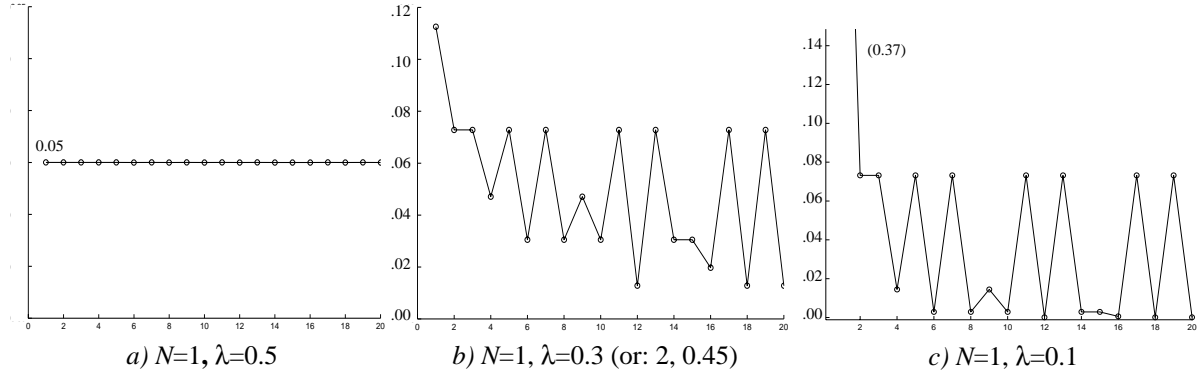


Figure 13. Sample selection probabilities for 20 harmonics, calculated using Algorithm 1.

How then to select good parameter values for N and the corresponding λ ? That depends on how rich the interfering polyphony of other sounds is. Some boundary values are illustrated in Figure 13. In plot *a*) one interfering sound is allowed to disturb half of the mass of $P_s(j)$, which results in an even distribution of selection probabilities. This is not sensible, since in that case even a single sound is able to disturb the output of the filter. On the other hand, in plot *c*) a single sound is allowed to disturb only 10 % of the mass of $P_s(j)$. This results in practically utilizing only the prime number harmonics of the observed sound. Plot *b*) gives a good tradeoff: one sound may corrupt 30 % of the mass of $P_s(j)$. We obtain an equal result by requiring that two sounds may corrupt 45 % of the mass of $P_s(j)$. It means that two interfering sounds should not corrupt the output of the filter, and still the distribution of $P_s(j)$ utilizes quite well the different harmonics of the observed sound.

The structure of P_s 's distribution is worth noticing. Our algorithm also solved a more general problem of finding values of vector $P_s(j)$ so that sums over every m^{th} value of it can be controlled. The solution is of the type that is plotted in Figure 13. In Figure 14, we further plotted $P_s(j)$

for 40 harmonics, and corresponding sums over every m^{th} harmonics.

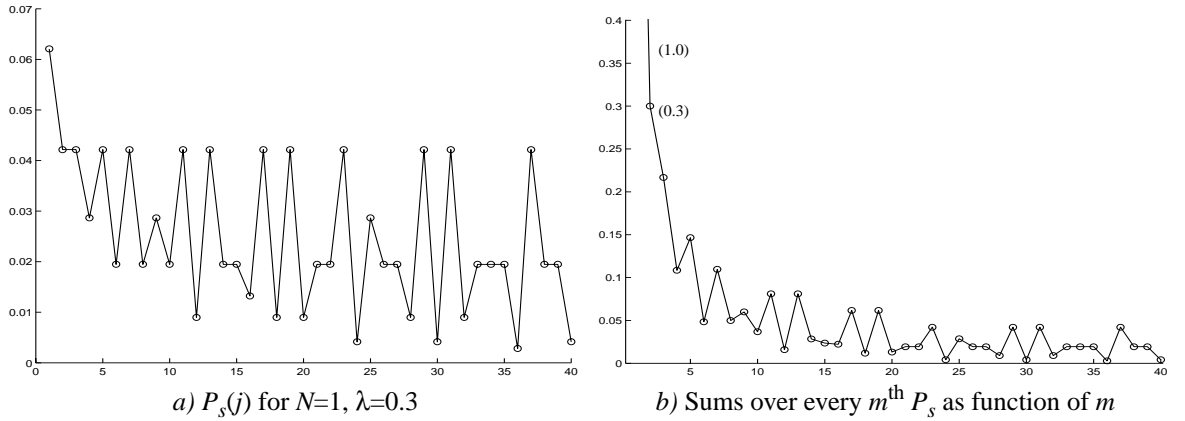


Figure 14. Sample selection probabilities for 40 harmonics and sums over every m^{th} value.

Example 3. Let us now reanalyse the *C major* chord that was used earlier in the Example 1 of Section 6.3. There we found out that 47%, 33% and 60% of the harmonics of the notes *C*, *E* and *G*, respectively, were overlapped by the other two notes in the chord. Now we study how large a percentage of the mass of the vector $P_s(j)$ is overlapped by the two other sounds when we change parameters in calculating $P_s(j)$. Setting λ to 0.4 we assure our filter to be robust for N interfering sounds (the remaining 0.6 will have a winning majority over 0.4). Then we let N increase and monitor $P_s(j)$. The results are in Table 8.

Table 8: C major. Percentage of overlapped P_s with different parameters.

Number of interfering sounds R_i that the filter ν is robust to	Parameters used in calculating P_s		Percentage mass of disturbed P_s (%)			Average
			<i>C</i>	<i>E</i>	<i>G</i>	
zero (old reference)	$N=1$	$\hat{\lambda}=0.5$	47	33	60	47
one	$N=1$	$\hat{\lambda}=0.4$	39	25	50	38
two	$N=2$	$\hat{\lambda}=0.4$	29	13	36	26
three	$N=3$	$\hat{\lambda}=0.4$	25	8	28	20
four	$N=4$	$\hat{\lambda}=0.4$	20	4	22	15

We note that there is a substantial difference between the old reference percentages and the filter which is robust to two interfering sounds, for example. Especially, we note that the overlapped amount can be effectively limited and controlled even in the worst cases.

We further reanalysed the other previously presented note combinations using only parameters $N=2$ and $\hat{\lambda}=0.4$ that makes our filter robust to two interfering sounds. The results are given in Table 9.

Two notifications should be made in Table 9. First, the disturbed percentages are substantially smaller using our filter than when equally utilizing all harmonics of the observed sound. Practically, the filter output will not be disturbed, if the percentage is 40 or smaller. Second, when *all* harmonics (100 %) are overlapped, nothing can be done, but the filter output is also disturbed. In this case we must use a so-called *subtraction principle* that will be presented in Section 6.9.

Table 9: Comparing the absolute overlap and overlapped mass of P_s .

Chord	Notes in the chord					Percentage of overlapped harmonics (%) , maximum bolded					Average	Percentage mass of disturbed P_s (%), maximum bolded					Average
<i>Cmajor</i>	c	e	g			47	33	60			47	29	13	36			26
<i>Cminor</i>	c	d#	g			33	20	50			34	18	13	26			19
<i>C⁰add6</i>	c	d#	f#	a		43	31	33	33		35	28	24	17	18		22
<i>C⁺</i>	c	e	g#	c2		60	25	30	100		54	36	10	17	100		41
<i>C⁹</i>	c	d	e	g	a#	50	23	39	65	11	38	31	8	17	40	5	20
<i>C+base</i>	c1	c3	e3	g3		47	100	100	100		87	26	100	100	100		82
<i>inharm.</i>	c	c#	d	d#	e	41	11	20	25	33	26	23	3	7	15	15	13

Emphasizing the lowest harmonics

In the beginning of this chapter we mentioned two defects to overcome. The other requirement, emphasizing the lowest harmonics, has not yet been treated. Above we set requirements (limits) to the sums over $P_s(j)$'s subsets E_m . Now we set further requirements for single $P_s(j)$ values. It turns out that further multiplying the values of $P_s(j)$ by a smooth and monotonically decreasing function does not substantially change the sums over each subset E_m , since subsets E_m are distributed over the whole length of $P_s(j)$.

The reasons that called for emphasizing the lowest harmonics were listed in the beginning of this section (page 37). Based on them, the emphasizing function $e(h_j)$ should be selected to roughly follow the general level of the harmonics of the sound whose features are observed. The selection of the emphasizing function is not critical, and the same emphasis can be used for all musical instrument sounds. One convenient function is given as an example:

$$e(j) = \frac{1}{j+a} - \frac{1}{J+a+1} , \quad (19)$$

where J is the set size. This function fits well for the relative loudnesses of the harmonics of different sounds, and can be easily tailored for the desired steepness by adjusting a . Setting a to zero reduces the envelope of the function to be $\frac{1}{j}$. For the observation of piano sounds we set a to J , which produces nearly linear roll-off.

Conclusion

We can now present the final sample selection probability vector $P_s(j)$ for a filter v that picks the estimated feature $\hat{g}(S)$ of a sound from the set of features $\{g(h_j)\}$ of its harmonics, and drops out the irrelevant values. It is a result of both sum constraints and an emphasizing function

$$P_s(j) = e(j) \cdot P_s^0(j) , \quad (20)$$

where $P_s^0(j)$ is the vector that was calculated using Algorithm 1 with parameters N and λ , and $e(j)$ is the emphasizing function. Recommendations for N and λ and $e(j)$ have been given. The sum over $P_s(j)$ must be scaled back to unity after multiplying by $e(j)$.

6.7 WOS filter to implement the desired sample selection probabilities

So far, we have fixed the probabilities of the harmonics to be selected to represent the observed sound. To benefit from that, we need to design a filter that *implements* these statistical properties when applied to the set $\{g(h_j)\}$. We present an iterative algorithm for finding a weighted order statistic (WOS, see Figure 12) filter to implement the selection probabilities, and to maintain the desired properties of a feature extraction filter that were listed on page 38.

As mentioned earlier, Kuosmanen has developed an algorithm for calculating the sample and rank selection probabilities of a WOS filter, whose weights and threshold T are known [Kuosmanen94]. There is, however, no analytic method to solve it the other way round: calculating the weights of a WOS filter to implement the desired sample selection probabilities. Thus an iterative algorithm was designed to meet the practical need in lack of an analytical method.

Algorithm 2. Calculation of the weights of a WOS filter to implement the desired sample selection probabilities $\{P_s(j)\}$.

Step 1. Initialize the filter's weights to be $w(j) = (P_s(j))^{0.75}$. After this, scale the overall sum of $w(j)$ to be a , which will be the size of the resulting multiset after weighting samples by taking $w(j)$ copies of each. The complexity of a WOS filter does not depend on a , but the complexity (and precision) of this algorithm does. A reasonable value for a is 100. If the number of input samples is over about 15, the approximation is precise enough and we can stop.

Step 2. Calculate the sample selection probabilities $\hat{P}_s(j)$ for the current weights of the filter using the algorithm presented by Kuosmanen in [Kuosmanen94, pages 41,42].

Step 3. Compare \hat{P}_s to the desired P_s . Increment $w(j)$ to the samples that have too low a selection probability and decrement $w(j)$ to the samples that have too high a selection probability.

Step 4. Scale the sum of $w(j)$ to be the selected size of the multiset, a .

Step 5. If the value of the vector $w(j)$ has changed and has not returned to an old, previously assigned value, return to step 2.

For a multiset size $a=100$, typically five iterations are enough for the algorithm to converge. A problematic step is the calculation of \hat{P}_s in step 2, whose memory consumption complexity is $O(2^J)$. This makes the algorithm heavy for sample set sizes larger than about 15. It turns out, however, that the initializing approximation $w(j) = (P_s(j))^{0.75}$ becomes more precise, when the sample set size grows. This makes further iterations useless for sample set sizes larger than about 15.

Threshold parameter T (see Figure 12) of the WOS filter must be biased using the knowledge of the statistic distribution between too-small and too-big outliers in data. In the later application to music transcription we used a median value, $T = \left\lfloor \frac{a}{2} \right\rfloor$, it is: choosing the middle value in the multiset of chosen size a . This decision was made based on the fact that imprecise instrument model produced most errors, making the distribution between too small and too big outliers quite equal. A growing number of interfering sounds adds too big outliers, as more harmonics are overlapped. In that case T should be selected smaller. Instead of selecting only the T^{th} value to the output, the mean of a group of samples around the T^{th} one can be used, too.

A filter was realized using the above presented procedure for sum parameters $N=2$ and $\lambda=0.4$, and sample emphasizing function $e(j)$ of Equation 19 with $a=J$. Its realized sample and rank selection probabilities for a sample set size 15 are plotted in Figure 15.

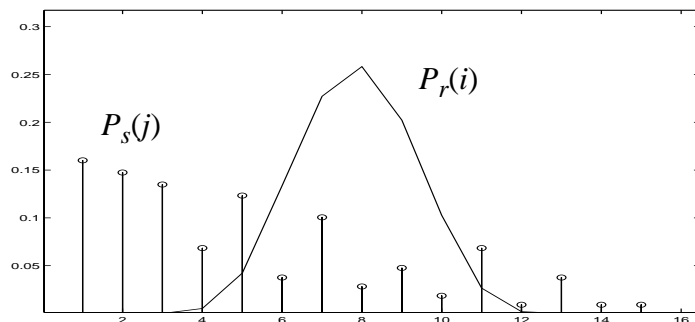


Figure 15. Sample (bars) and rank (curve) selection probabilities of an example WOS filter.

Practical use of the WOS filter

We reduced the observation of a feature $g(S)$ of a harmonic sound S in the presence of other interfering harmonic sounds R_i to measuring the features $g(h_j)$ of the harmonics of the sound S and applying a weighted order statistic filter v to deduce an estimate for $g(S)$.

The result of the whole design process described in this chapter can be packed to a small kernel of filters, designed to apply for the feature extraction of different kinds of sounds. A good starting point is using the recommended selections for N , λ and $e(h_j)$. WOS filter weights must be stored separately for each different sample set size, but this is not a problem, because the typical set sizes J , i.e., the amounts of audible harmonics of sounds, go approximately up to fifty, and the amount of data stored for each filter is about J bytes.

A ready-made filter kernel can be effectively used, but its optimal design calls for understanding the presented theoretical grounds along the design path. A proper understanding of them and the knowledge of the statistical properties of the data opens a way to tailor an optimum solution for each feature extraction problem.

6.8 Extracting features that cannot be associated with a single harmonic

In the Section 6.1, where the feature operator $g(\bullet)$ was defined, we divided the features into two classes, depending on whether they can be associated with a single harmonic or not. Until now, only features of the first class have been considered, since we have applied our feature extraction filter v to the set of features of the harmonics $\{g(h_j)\}$. For the feature class II, however, the notation $g(h_j)$ is irrelevant, because g cannot be applied to a single harmonic.

Again we take sound *colour* $g_C(S)$ as an example of a class II feature. The perceived sound colour of a harmonic sound at a time instant t_0 is determined by the relative amplitudes of the harmonics, when compared to each other, i.e., the energy distribution of the sound to its harmonics. Such a feature cannot be associated with a single sinusoid.

The kernel of the designed filter v can still be used in the extraction of class II features. We first need to define a *parametrizable* model for the sound colour, where the number of parameters is less than the number of harmonics. Let us estimate the relative amplitudes of the harmonics with, say, a second order polynomial $g_A(h_j) = aj^2 + bj + c$.

Now extracting this model's parameters from the observed signal in a manner analogous to the WOS filtering is an optimization problem. We need to determine the parameters a , b and c so that L1 error, i.e., the absolute value of the difference between the model's amplitude values and the values of the actual harmonics in the spectrum is minimized. Further, the L1-error over the sample points must be taken as a *weighted* sum, where the weight of each sample is according to the sample selection probabilities $P_s(j)$. The meaning of the threshold T of the filter can be taken into account by giving a different emphasis to the positive and negative deviations from the actual sample values. Fitting a function to datapoints in L1 sense is quite a usual optimization problem, and is not treated here.

Adaptive instrument modeling is a problem to be solved using this kind of procedure. It means the problem of finding the colour of a tone (sound of a musical instrument) from the midst of a polyphonic piece of music by adapting the model's parameters along with the playing of the piece.

6.9 Feature 'subtraction' principle

Our algorithm and discussion on the observation of the features of sounds in the presence of other harmonic sounds was based on an assumption that the observed sound S is not *totally* overlapped by an interfering sound R , whose fundamental frequency is

$$f0_R = \frac{1}{n} \cdot f0_S$$

As a last case, we will now propose a method for dishing up this worst case. To repeat: the observed sound is totally buried under the harmonics of an interfering sound, and even this is not an extremely special situation, because of the characteristics of music that were discussed in Section 6.3.

The basic idea of our solution is to *compensate* the effect of the interfering sound R , the properties of which can be robustly extracted in the presence of S using the procedure presented before, because the interfering sound is not totally overlapped by S . Thus it will be enough to develop an algorithm to subtract, remove, or compensate, the revealed properties of the lower sound and then proceed to determine the properties of the sound S , which is laid bare from under the interfering sound.

The subtraction process crucially depends on the feature under inspection, and cannot be presented in a general form. Instead, we treat one example that has often been discussed in the field of automatic transcription of polyphonic music [Chafe85, Brown91b, Martin96a,b].

Example 4. The problem is to determine the hypothetical *existence* and *loudness* of a sound S at the fundamental frequency $f0_S$, which has been totally overlapped by a lower sound, whose fundamental frequency is in the aforementioned $f0_R = \frac{1}{n} \cdot f0_S$ relation to S .

The existence of the lower sound R can be deduced using the methods presented in this chapter. Moreover, its loudness $g_L(R)$ is robustly determined in spite of every n^{th} harmonic being overlapped by the sound S . But for the sound S it is most difficult to determine whether it even exists (not to speak about recovering its loudness), because all its harmonics are overlapped by the harmonics of the sound R and could therefore be resulted from the sound R alone. The existence of the sound S is indeed theoretically ambiguous because the same signal content could really have been resulted from a single

harmonic sound having stronger harmonics at every n^{th} positions.

There is, however, a remedy to the problem of detecting the higher one of the two sounds in a harmonic relation to each other. The higher sound is revealed by the *regularity* at which the amplitudes of the harmonics at every n^{th} positions exceed the overall level of the harmonics of R . Although this cue is theoretically ambiguous, as stated above, it is both psychoacoustically relevant and practically useful.

The principal problem in subtracting the harmonics of R from the spectrum is that although an estimate of the feature $g_L(R)$ can be deduced from the features $\{g_L(h_j)\}$ of its harmonics, this cannot be done vice versa, i.e., $\{g_L(h_j)\}$ cannot be precisely deduced from $g_L(R)$, although it would be desirable to enable subtracting the loudness of each single harmonic from the corresponding sinusoid in the spectrum.

We approximate subtracting, however. Some parametrizable curve is fitted in L1 sense through the spectrum peaks at the slots of the harmonic partials of R . This is done using the procedure presented in Section 6.8. Now this curve determines the approximated distribution of the robustly calculated loudness of R to its harmonics. It turns out that the approximation made in the loudness distribution does not crucially change the loudness of S , when determined after the subtraction. Subtracting a set of sinusoids from a Hamming windowed Fourier transformed spectrum is another story altogether, and will be explained, when our automatic music transcriber is considered in the next chapter.

After the sinusoid partials of the sound R are removed, we notice that a fundamental frequency candidate of S at the position of the n^{th} harmonic of R stands naked in the spectrum. When its loudness is determined using the same procedure as for R , it turns out that the sound S truly exists, because it is loud enough to be heard as an independent sound. It is seen that a robust determination of the loudness of the lower sound gives a steady ground to approach even totally overlapping set of potential sound candidates in a signal by starting from the lowest one, determining its loudness, subtracting it, and continuing upwards to the next candidate.

7 Applying the New Method to the Automatic Transcription of Polyphonic Music

In this chapter we evaluate and verify the theoretical results of the previous chapter by applying them in a computer program whose purpose is to transcribe polyphonic piano music.

The challenge to our system is shortly stated as follows. We take the instrument to be transcribed and record its different notes one by one, a sufficient amount to represent all the different *tone colours* that can be produced by that instrument. In the case of a piano we recorded all its keys one at a time. This is what we call *training material*. When our program has studied the training material we require that it can transcribe rich polyphonic musical signals played with the same instrument, i.e., to decompose signals into a subset of training set tones and to determine their loudnesses and fundamental frequencies in the signal. Transcribing without training material will be discussed, too.

This is a good simulation arrangement to evaluate the efficiency of the presented theoretical methods in resolving a mixture of harmonic sounds. Emphasis is laid on resolving *rich polyphonic contents* that have been previously impossible to transcribe. We will present simulation results both for ‘typical stumbling point’ note mixtures and for classic piano compositions. We will also compare this transcription system to our older pattern recognition approach, reflect their differences and strengths, and evaluate the role of the new methods.

7.1 Approach and design philosophy

Music transcription is a complicated cognitive task and requires integration of various forms of knowledge. A human listener not only uses the information that is present in the signal, but also utilizes his prior knowledge about musical conventions and instrument tones. Here we use a so-called bottom-up approach only, which means we ignore musical knowledge and only use the information that can be found from the recorded signals. In other words, all note combinations and sequences are assumed equally probable.

In Section 3.6 we presented a certain design philosophy, which goes through the design of the system as a background flow of thought. Shortly, at each abstraction level we must first know what information is needed, then find the best method to extract it, and finally analyse the reliability of the measurements. This philosophy has by far determined the selection of system building blocks from among the tools presented in literature. An additional objective was to try to concentrate on algorithm design and to avoid spending time on parameter tuning.

7.2 Overview of the system

Overview of the system is presented in Figure 16. It consists of three processes, where two supporting processes pass their results to a *transcription process* in the middle. Output from studying the training material is a bank of tone models, that is, the parametrized information of the tones that may exist in signals that should be transcribed. The studying process is hereafter called *tone model creation process*. It makes a radical assumption that only one musical tone plays at a time in the training material and then proceeds by segmenting the signal and deter-

mining the fundamental frequency and tone colour of each musical sound. The other supporting process, *kernel creation process* is an implementation of the algorithms presented in the previous chapter. It yields filter kernels that are used in observing harmonic sounds in the transcription process.

The kernel creation process needs to be run only once, the tone model creation process needs to be run once for each new instrument, and the transcription process needs to be run for each musical piece to be transcribed.

The two signal sources at the bottom of Figure 16 stand for the training material and a musical recording to be transcribed. The third input, facts concerning sound mixtures, refers to the characteristics of a mixture of harmonic sounds that were discussed in the previous chapter.

7.3 Filter kernel creation process

The filter kernel creation process is an implementation of the algorithms that were presented and motivated in the previous chapter. Its aim is to find a feature extraction filter to observe harmonic sounds in the presence of interfering sounds. Algorithm 1 was programmed to calculate sample selection probabilities $P_s(j)$ that are robust to two interfering sounds in the worst

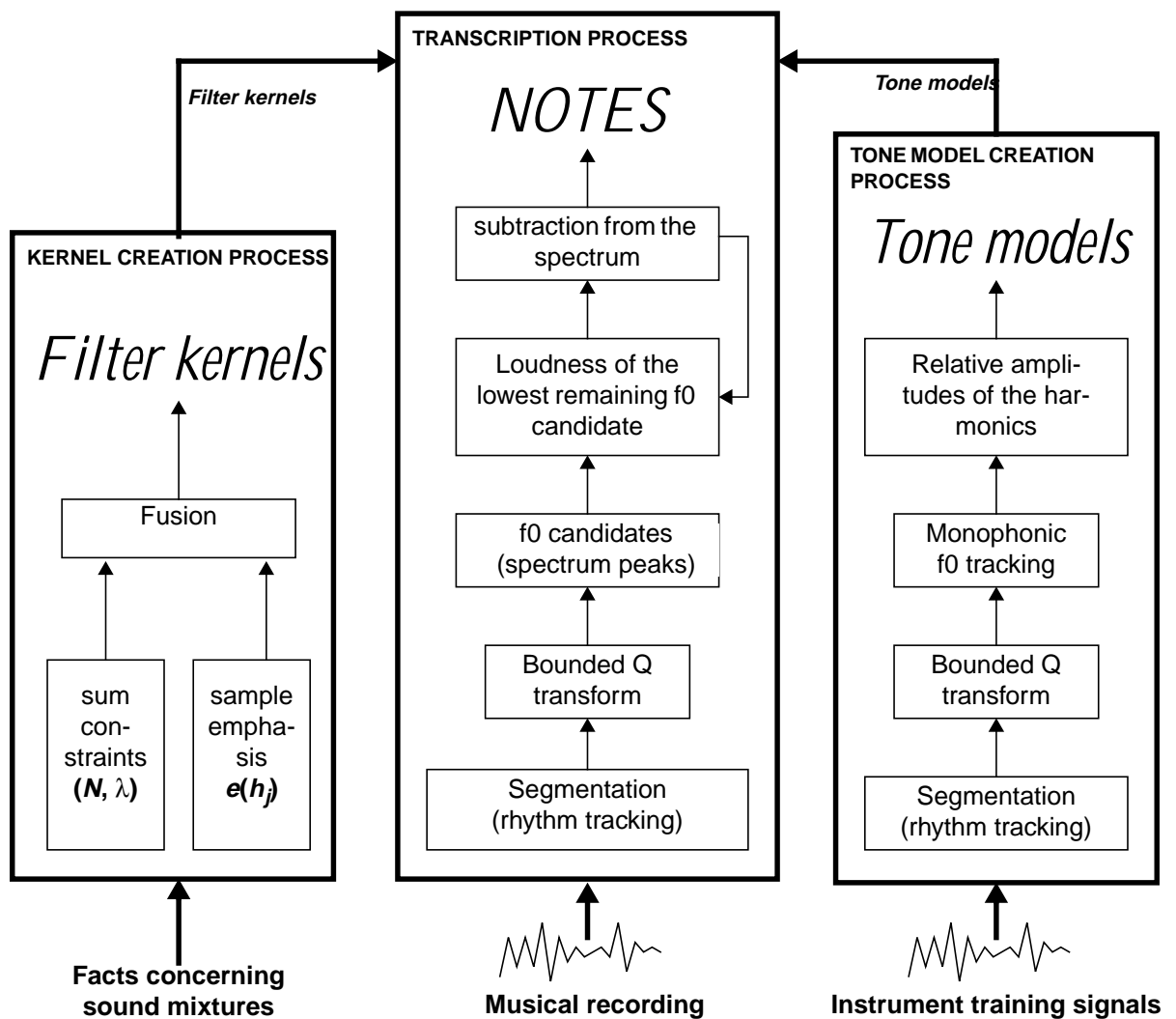


Figure 16. Overview of the system.

case ($N=2$, $\lambda=0.4$). The lowest harmonics were emphasized using the recommended function on page 43 with $a=J$. Algorithm 2 is used to find a weighted order statistic (WOS) filter, which realizes the desired sample selection probabilities. The threshold T was set to pick the median value among the representatives in order to effectively drop out both too small and too large outlier values.

The WOS filter parameters, kernels, must be calculated for all sample set sizes from 1 up to the maximum number of harmonics that is expected to be encountered in the observed sounds. The results of this process are stored in a data structure, kernel of filters, which is then used in observing the features of the sounds in the transcription process.

7.4 Tone model process

Input to the tone model process is a set of monophonic training signals, where each instrument is played at a sufficient number of pitches and styles to represent the whole range of *tone colours* that can be produced by these instruments. The process makes a radical assumption that only one musical sound plays at a time in training signals. Then it makes a monophonic transcription of signals by segmenting them, observing the sounds, and storing the parameters of the sounds to tone models.

The information stored for each tone is its fundamental frequency and relative amplitudes of the harmonics. Measurements are focused to a time instant at about 30 ms from the tone onset, where most transient phenomena have subsided, but the higher harmonics are still playing, and the whole richness of the harmonic series can be utilized in calculations. As discussed in Chapter 3, the bounded Q transform was chosen for the frequency analysis to provide precision both in time and logarithmic frequency. This allows focusing the transform to a certain time instant or observing the time evolution of the sound. However, in the current system the transform and the measurements are focused to one time point only. This is a gross simplification, since even a human listener usually cannot recognize a sound without hearing its evolution in time and the beginning transient. This simplification is justified, however, by our intention to focus on evaluating the performance of the algorithms presented in the previous chapter. Using a better, time evolving model will be discussed later in this chapter, and more extensively in Section 8.4.

The inner structure of the tone model process is presented in Figure 17. Main control is on the top, and it proceeds by calling the subroutines in succession from left to right. Text “for each *<item>*” on the calling arrow means that the call is repeated for each new *<item>*. The subroutines in turn may call lower level subroutines.

The main control proceeds roughly as follows. Each training signal is loaded at a time and segmented to distinct tones using the rhythm tracking procedure that was presented in Chapter 4. The segments are bounded Q transformed one by one, and the fundamental frequency and the relative amplitudes of the harmonics of the tone are solved from the spectrum. Parameters of the tone are then added to a tone model bank. In the following sections we describe the different phases of this process in the order of the program’s flow of control.

Bounded Q transform

The principles of the bounded Q transform were presented in Chapter 3. It was chosen for our system because of its ability to efficiently calculate a spectrum with a logarithmic frequency resolution. In our implementation three parameters can be tuned: 1) the length of the shortest

time domain window (for the highest frequencies), 2) the number of octaves to iterate for low frequencies, doubling the frequency precision and time window for each octave, and 3) the point in time where the frequency analysis is focused and the time domain windows are centered to. These analysis parameters should be selected to be the same both in the tone model creation and transcription processes to optimize their co-operation.

A Hamming window is used in frequency transforming each time domain window. This is important, because a Hamming window prevents a single sinusoid from blurring too wide in the frequency spectrum, which in turn allows a more reliable *sharpening* of the spectrum peaks. The sharpening operation will be described later.

Tracking the tone onsets

The scheme in tracking the onset times of the tones is the one presented in Chapter 4. As stated earlier, too sensitive tracking is not a problem here, since two rhythmic segments can be later fused, if the latter segment is noted to contain only continuation of earlier started sounds.

Monophonic fundamental frequency determination

Monophonic fundamental frequency determination is not trivial, but still easy from our point of view. Roughly, it is done by forming a set of f_0 candidates from the lowest spectrum peak and two and three times lower frequencies below it. Then each harmonic partial above a candidate is taken as a piece of evidence for the existence of the candidate, emphasizing the harmonics according to the weights $P_s(j)$ that are calculated using Algorithm 1 of the last chapter and

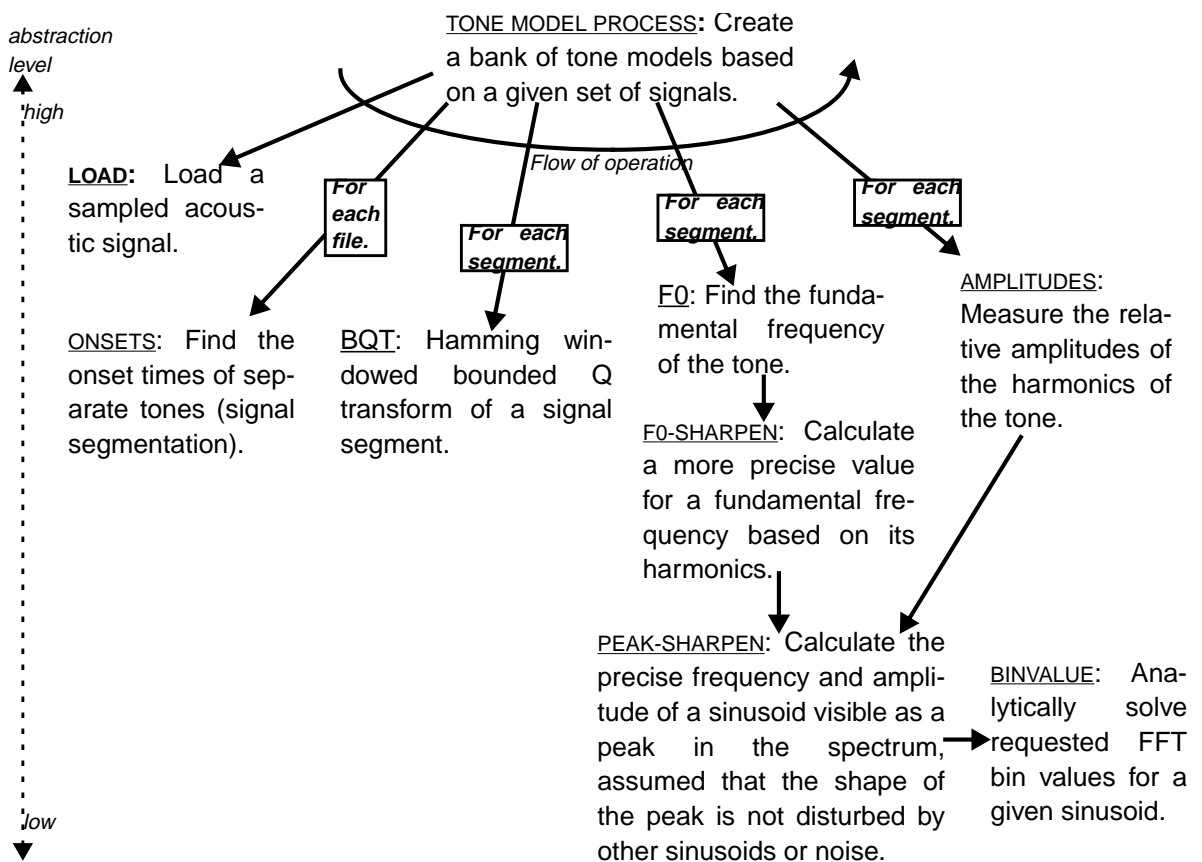


Figure 17. Tone model creation process: subroutines and their calling relations.

the same parameters as in the kernel creation process. The winning f_0 candidate is chosen.

Analytical calculation of a Hamming windowed Fourier transform of a single sinusoid

In the following two sections we present mathematical tools that bridge the gap between the *sinusoid* representation of a sound and its *spectrum* calculated using a Hamming window (see Figure 18). First we derive a result that allows efficient analytical calculation of the desired frequency domain bins for a known sinusoid.

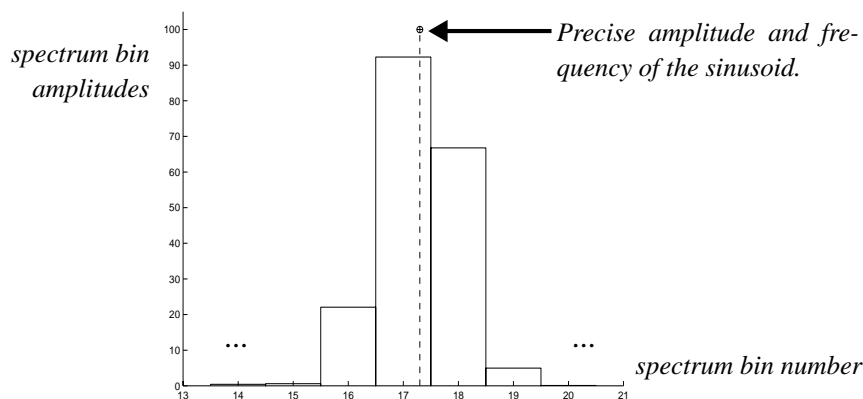


Figure 18. Hamming windowed frequency transform of a single sinusoid.

Discrete Fourier transform $X(k)$ of a sampled time domain signal $x(nT)$ using a Hamming window is calculated as

$$X(k) = \sum_{n=0}^{N-1} \left(x(nT) \cdot e^{-j \cdot k \cdot \frac{2\pi nT}{N}} \cdot \left(0.54 - 0.46 \cdot \cos\left(\frac{2\pi nT}{N-1}\right) \right) \right), \quad (21)$$

where j is the imaginary unit, T is the sampling interval, and k is the frequency bin number, zero corresponding to zero frequency, and $k = \frac{N}{2}$ to half the sampling rate [Ifeachor93].

A harmonic sound consists of a series of sinusoids. A stable time domain sinusoid in the time domain window $[0, N-1]$ can be expressed as

$$x(nT) = A \cdot e^{j \cdot f_s \cdot \frac{2\pi nT}{N}}, \quad (22)$$

where A and f_s are the amplitude and frequency of the sinusoid, and n goes from zero to $N-1$. Here we deliberately omit the phase of the signal, since it is not used. Substituting (22) to (21) we can calculate the value of a transform bin k for a single sinusoid as

$$X(k) = \sum_{n=0}^{N-1} \left(A \cdot e^{j \cdot (f_s - f_k) \cdot \frac{2\pi nT}{N}} \cdot \left(0.54 - 0.46 \cdot \cos\left(\frac{2\pi nT}{N-1}\right) \right) \right), \quad (23)$$

where f_k is the frequency of a spectrum bin k . This sum approaches the value of a continuous integral as the sampling interval T approaches zero. Therefore we can approximate the summation by replacing it by a continuous integral, writing t instead of nT , and integrating t from zero to $N-1$. The integral can be analytically solved, and yields

$$\int_0^{N-1} A \cdot \left(0.54 \left(-j \cdot e^{\frac{j \cdot \beta \cdot t}{\beta}} \right) - \left(0.46 \left(\frac{1}{2} \cdot \left(\frac{j \cdot e^{-j \cdot (\alpha - \beta) \cdot t}}{\alpha - \beta} + \frac{e^{j \cdot (\alpha + \beta) \cdot t}}{\alpha + \beta} \right) \right) \right) \right), \quad (24)$$

where $\alpha = \frac{2\pi}{N-1}$ and $\left(\beta = 2\pi \cdot \left(\frac{f_s - f_k}{\text{samplingrate}} \right) \right)$.

This means that we can effectively calculate the required bins of a Hamming windowed Fourier transform of a single sinusoid by substituting the upper and lower bounds of t to Equation . This result will be later used extensively when subtracting known sinusoids from the spectrum in the transcription process.

Sharpening a spectrum peak

Next we present a tool for doing the previous operation backwards, i.e., calculating the precise frequency and amplitude of a sinusoid based on spectrum bin amplitudes around a spectrum peak, as presented in Figure 18.

Often we do not have too much redundant information in musical signals, but the notes play just long enough to determine the frequencies of the sinusoid partials that make up sounds. Low notes typically play longer in music, because even the human ear is governed by the laws of the nature and needs a longer time period to distinguish sinusoids that are low and close to each other in frequency. However, we do not have arbitrarily long time periods to find their exact frequencies, although that would benefit us a lot. Thus we need to use the available time window length, and later deduce more precise parameters of the sinusoids, if possible.

Consider a single sinusoid in a noise-free signal. If we could solve A and f_s from Equation as a function of two known spectrum bin amplitudes, we would be able to calculate the exact frequency and amplitude of a sinusoid even in a quite short period of time, based on the *shape* of a spectrum peak which indicates the sinusoid. It is evident that our signals of interest are not noise-free and contain several sinusoids. That is not fatal, however. First of all, when a Hamming window is used, interfering sinusoids do not spread very wide in the spectrum, and the most important sinusoids are not typically close to each other, since the lowest harmonics have a large interval between each other in a logarithmic scale. Second, when we use the two strongest spectrum bins near the peak, the effect of noise does not prevent us from correcting the frequency of the sinusoid still a lot when compared to just picking the highest spectrum bin.

We could not solve A and f_s analytically from Equation , but an approximation was derived. Thus we wrote a procedure which takes as input the amplitudes of the bins of a spectrum peak and the two bins around it, and calculates the frequency and amplitude of a stable sinusoid, which causes a peak of such shape, see Figure 18. Although noise and interference was ignored and an approximation was used, this analysis makes an important positive contribution to the results of the higher level algorithms that use the parameters of the sinusoids as their input.

Precise values of the fundamental frequency and of the amplitudes of the harmonics

To find a precise fundamental frequency value of a hypothetical sound candidate we utilize the whole series of its harmonics. The procedure is written in the form of an algorithm.

Algorithm 3. Finding a more precise value for the fundamental frequency of a sound.

Step 1. Look at the positions in the spectrum that are multiples of an initial, still imprecise fundamental frequency of the sound. Form the set $\{h_j\}$ which includes only the harmonics that appear as *peaks*, roughly in these spectrum positions.

Step 2. Sharpen the peaks $\{h_j\}$ using the procedure presented in the previous section and form a set of sharpened frequencies of the harmonics $\{g_F(h_j)\}$.

Step 3. Calculate a *weighted mean* among the frequencies $\left\{ \frac{g_F(h_j)}{j} \right\}$ using as weights the selection probabilities P_s of the feature extraction filter ν that was designed in the previous chapter. Weights are zero for the harmonics that are not peaks in the spectrum.

A weighted mean is used instead of a WOS filter, because there cannot be radically disturbed outlier values in the set $\{g_F(h_j)\}$, and, on the other hand, no single frequency bin is likely to be good enough to be picked alone as a representative for the fundamental frequency.

The amplitudes of the harmonics of the tone are found by looking at the positions of the harmonics and picking the bin amplitudes. If a bin appears as a peak, i.e., the bins at both sides have a lower amplitude, the peak is sharpened using the procedure presented in the previous section, and the refined amplitude value is used.

7.5 Transcription process

Input to the transcription process is a musical signal that should be transcribed, together with tone models and filter kernels from the other two processes (see page 49). The flow of operation goes roughly by rhythm tracking the note onset times in music, and then resolving the contents of each signal segment one by one. The *output* of the process consists of 1) segment times, which imply the onset times of the notes and 2) of the fundamental frequencies, loudnesses and instrument types of the sounds in each segment. The fundamental frequencies are not quantized to those of tone models, but the closest (in frequency) tone model is used for each instrument.

The inner structure of the transcription process is presented in Figure 17. The notation is the same as was used before. Moreover, some of the procedures are the same as were used in the tone model creation process, and are not explained again.

The very heart of the process is the subroutine RESOLVE TONES, which decomposes a signal segment to a set of tones and finds their fundamental frequencies and loudnesses. It proceeds roughly as follows. First, all potential *fundamental frequency candidates*, also called *sound candidates*, are deduced from the spectrum. Then the candidates are treated in an ascending frequency order by resolving their loudnesses and instrument types, subtracting them from the spectrum and moving on to the next candidate. Sounds that truly exist in a signal should acquire a clear prominence in loudness, and therefore the loudest sounds are selected to output.

From now on we consistently denote sound candidates by C_i , tone models by T_i , and truly existing sounds by S_i . The only parameter of a sound candidate is its fundamental frequency, this is why it is also called a fundamental frequency candidate. Tone models and observed sounds also have the parameters loudness and instrument type.

Finding fundamental frequency candidates from the spectrum

The spectrum of a rich polyphonic musical signal may seem quite inextricable at a first glance, and indeed, *there is no trivial way of choosing the potential sound candidates C_i* . All spectrum peaks must be involved, and even this is not enough, but some more C_i must be added, not to ignore any truly existing sounds. As an example, look at the bounded Q transform of one rhythm segment from Mozart's Rondo alla turca in Figure 20. The circles denote the fundamental frequency candidates that were picked using the algorithm below.

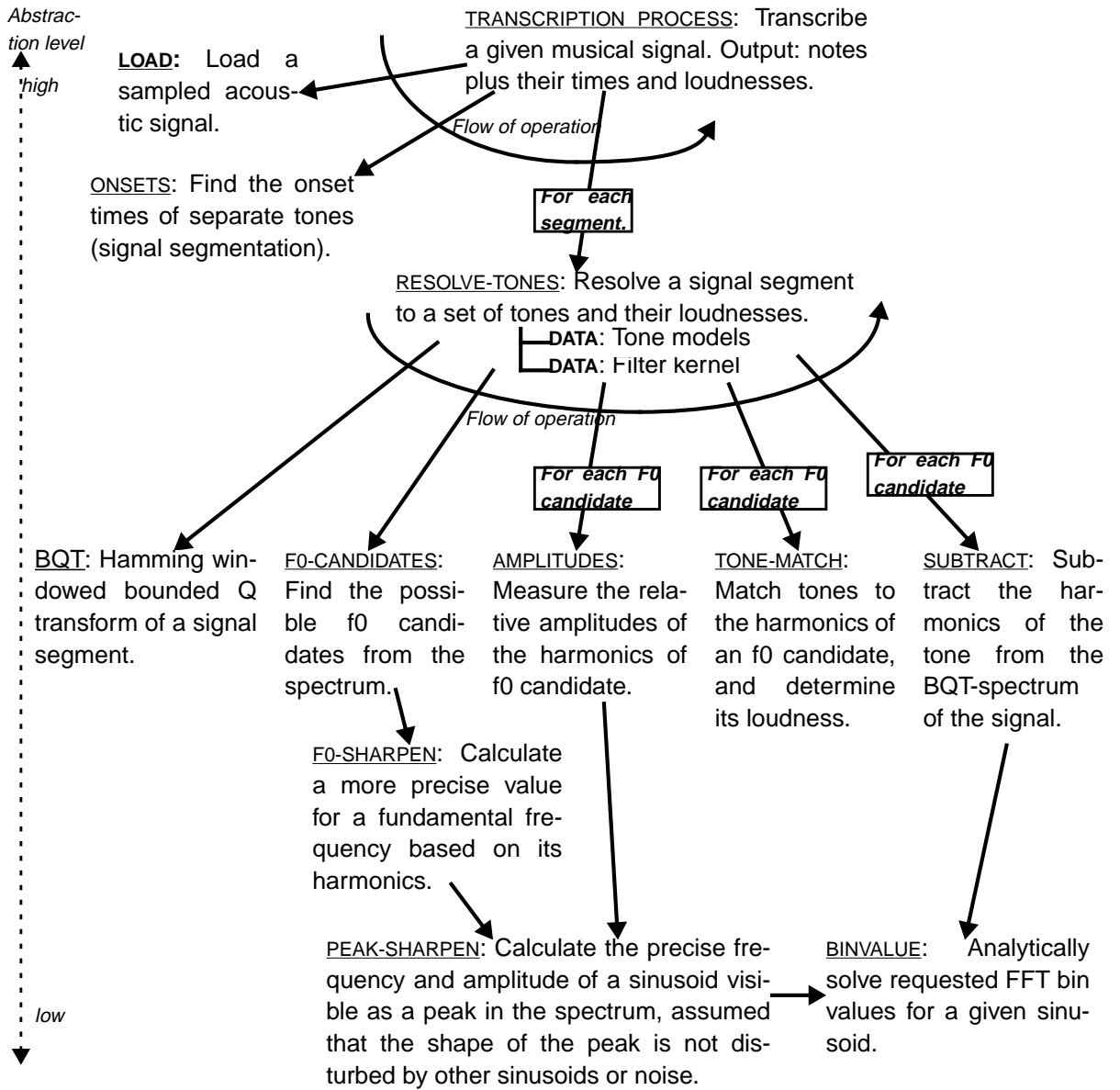


Figure 19. Tone model creation process: subroutines and their calling relations.

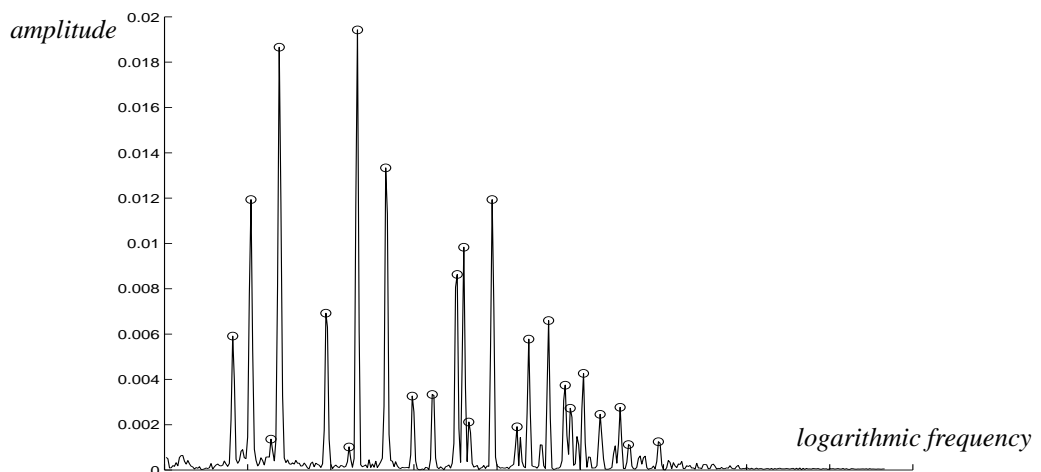


Figure 20. Bounded Q transform spectrum of one rhythmic segment in Mozart's Rondo alla turca. The fundamental frequency candidates are circled.

Algorithm 4. Finding potential fundamental frequency candidates C_i from signal spectrum. Input parameters are: *a*) Minimum possible frequency interval between two fundamental frequencies, $\Delta f0_{min}$. In music this is the distance between two adjacent notes, and is

$$\Delta f0_{min} = \frac{f0_1}{f0_2} = 2^{\frac{1}{12}}.$$

b) Minimum acceptable loudness of a fundamental harmonic partial. The Algorithm proceeds as follows:

Step 1. Collect all peaks from the spectrum, i.e., all frequency bins where the adjacent bins at both sides have a smaller amplitude than the middle one.

Step 2. Drop out the candidates whose loudness falls below the minimum acceptable value, and the candidates that are too close to a stronger candidate.

Step 3. Find the precise frequency values for the peaks by utilizing the peak sharpening procedure that was presented earlier.

Step 4. *Widen* the peaks by adding new members to set $\{C_i\}$ at $\Delta f0_{min}$ frequency steps around the peaks as far as the bin amplitudes are decreasing and the amplitudes of the bins are high enough to be accepted. This widening is needed not to ignore true $f0$ s that are masked by a close lying and higher amplitude frequency peak.

Step 5. (Optional.) If very low sounds are considered, we must generate C_i that are two times lower than the lowest candidates, because the fundamental harmonic of low sounds may be totally missing.

Finding more precise values for the frequency of the candidates may seem to be needless extra refining, but it turned out that it has a remarkable effect on the chain of operations to follow.

Matching tone models to find *the loudness and tone type of the candidates*

The inspection of C_i must go in an ascending frequency order to avoid supposing a harmonic of a sound to be a fundamental frequency, because all the ‘harmonics’ of a harmonic really exist in the spectrum. This error is avoided by a robust determination of the loudness of the lowest remaining C_i , subtracting it, and continuing upwards to the next candidate. If a harmonic of a truly existing sound is not another true fundamental, its ‘harmonics’ have been subtracted away, and that mistake will be made no more.

Let us now present an algorithm which solves the loudness and tone type (instrument) of a sound candidate C . To do that we need to define a new feature $g_W(T_i, C)$, which represents the *weight*, or more exactly, the fit of a tone model T_i to signal spectrum at the positions of the harmonic series of a sound candidate C . Feature g_W can be easily defined for a single harmonic as

$$g_W(h_j, T_i, C) = \frac{g_A(h_j, C)}{g_A(h_j, T_i)}, \quad (25)$$

where $g_A(h_j, C)$ is the amplitude of the harmonic h_j of the candidate, picked from its position in the spectrum in the same manner as in the tone model creation process, and $g_A(h_j, T_i)$ the amplitude of the corresponding harmonic in the tone model.

In an ideal case, if the tone is alone in the signal, noise is ignored, and the tone model is perfect, $g_W(h_j, T_i, C)$ is the same for all h_j and determines the weight of this particular instance of the tone in signal. These conditions are, however, not valid in true polyphonic musical signals,

especially because interfering sounds overlap subsets of every m^{th} harmonics of the candidate, as discussed earlier. Thus we need to calculate the weight of a whole tone model T_i in candidate C as

$$g_W(T_i, C) = v\{g_W(h_j, T_i, C)\} , \quad (26)$$

where $v(\bullet)$ is the filter that was designed in the previous chapter, and $\{g_W(h_j, T_i, C)\}$ is the set of weights of the harmonics of this tone. This value is trustworthy, because we know that there is no sound in a straight harmonic relation *below* the current candidate, or if there is, it has been subtracted already. On the other hand, only a partly disturbed set $\{g_W(\bullet)\}$ is handled by the properties of the filter v .

The *loudness* of a tone model with unity match can be determined by summing the loudnesses of its harmonics, calculated by substituting their frequencies and amplitudes to *normal equal-loudness level contours* [ISO87]. A rough reference value for the loudness of a tone model can also be calculated as

$$g_L(T) = a \cdot \sum g_F(h_j, T) \cdot g_A(h_j, T) , \quad (27)$$

where a is a constant, $g_F(\bullet)$ means frequency, $g_A(\bullet)$ amplitude. This rule-of-thumb works quite well if the harmonics h_j of the tone are in the frequency range 20...2000 Hz. However, using normal equal-loudness level contours is highly recommended. The loudnesses of the tone models need to be calculated only once, since the loudness of a candidate sound C_i can be calculated by finding the best matching tone model T_{best} for it, and scaling the loudness of T_{best} according to its fit $g_F(T_{best}, C_i)$ to C_i . Thus the loudness of a candidate C_i can be written as

$$g_L(C_i) = a \cdot g_W(T_{best}, C_i) \cdot g_L(T_{best}) . \quad (28)$$

The above discussion is rewritten in a form of an algorithm.

Algorithm 5. Finding the tone type (instrument) and loudness of a fundamental frequency candidate C .

Step 1. For each instrument, find the tone model T_i , whose fundamental frequency is closest to the fundamental frequency of C .

Step 2. Match tone models to C by calculating their fits $g_W(T_i, C)$ to C .

Step 3. Calculate the loudnesses of C for the different T_i by substituting their fits and unity loudnesses to Equation 28.

Step 4. Choose the instrument, which gives maximum loudness to C , and thus indicates to be the best matched instrument. Choose the corresponding loudness to be the loudness of C .

The fundamental frequency of the candidate is already known, and thus we have also found the other parameters of the candidate.

Subtraction procedure

In music it quite often happens that two notes are in a

$$f0_{lower} = \frac{1}{n} \cdot f0_{higher} \quad (29)$$

relation to each other, and n is an integer number. A method to make correct observations of the lower sound were earlier presented, but to detect and observe the higher sound we need a

method to lay it bare from under the lower sound which totally overlaps it. This method was presented earlier, too, and was named subtraction principle.

The features of concern in our transcription program are weight $g_W(T_i, C)$ and loudness $g_L(C)$, which are both based on the *amplitudes* of the harmonics. Thus the very way to compensate the effect of the lower sound is to subtract the amplitudes of its harmonics from the spectrum. There is no trivial way of subtracting a known sinusoid from spectrum. Instead, the following procedure must be gone through.

Algorithm 6. Subtracting harmonic partials of a sound C from a spectrum, when its tone model T_i and weight of the model $g_W(T_i, C)$ are known.

Step 1. Initialize j to 1.

Step 2. Calculate the frequency $g_F(h_j)$ of harmonic j by

$$g_F(h_j) = j \cdot g_F(C)$$

and the amplitude $g_A(h_j)$ of harmonic j by

$$g_A(h_j) = g_W(T_i, C) \cdot g_A(h_j, T_i)$$

where $g_A(h_j, T_i)$ is the amplitude of the corresponding harmonic in the tone model.

Step 3. Form a set of spectrum frequency bins $\{b_k\}$, that are close enough to frequency $g_F(h_j)$ so that their amplitude value has been changed by the harmonic h_j . The size of the set $\{b_k\}$ depends on the windowing function that was used in the frequency transform, since the windowing crucially affects how far the shape of a pure sinusoid h_j blurs in the spectrum. For a Hamming window it is enough to include two or three bins at both sides of the frequency $g_F(h_j)$.

Step 4. Analytically solve the frequency transform $\{H_i\}$ of the harmonic h_j to the spectrum bins $\{b_j\}$ using the procedure that was presented earlier in this chapter. This is the amount of amplitude that the harmonic partial h_j has brought to these bins.

Step 5. Subtract $a \cdot \{H_i\}$ from the corresponding bins $\{b_i\}$ to remove the sinusoid from the spectrum. The value of the constant a is discussed right after describing this algorithm.

Step 6. If j is smaller than the highest audible harmonic of C , add j by one and return to step 2.

What is the right value for the constant a ? When we sum two sinusoids having amplitudes A_1 and A_2 and the same frequency, the amplitude of their sum depends on the phase difference of the two sinusoids, being between zero and (A_1+A_2) . Value 1 for a is motivated by the fact that we cannot know if we have other sinusoids under the ones that are being subtracted, or not. Still we must be sure to totally remove the harmonics that are not overlapping anything. Consider a sound C which is alone in the signal. If we subtract using a value 0.64, for example, the next candidate (the second harmonic of C) will receive a large fraction of the energy of C , and will erroneously be taken as an independent sound. It is thus safer to subtract enough, even though it reduces the amplitude of the sounds that are totally overlapped by the subtracted one. For other sounds it is not fatal to slightly corrupt the values of every k^{th} harmonic ($k>1$), because of the properties of filter v .

Segregation between true and prune sound candidates

Selection of the truly existing sounds from among the candidates is based purely on their loudnesses. To adapt to the overall signal level, the loudnesses are scaled so that the loudness of the most prominent sound is unity. Candidates whose relative loudnesses are over a certain threshold are then accepted as true sounds. This is an extremely simple model of the *masking* phenomena of the human ear: sounds with too low a relative loudness are masked by the louder sounds, and are not heard. The relative loudness 0.2 is a good starting point for the threshold.

This loudness-based segregation is straightforward and simple, but works quite well, though, as can be seen from the simulation results. We consider spending too much time on setting thresholds a waste of time, because the very basis of the calculations is in the preceding algorithms, which should give clear prominence to the right sounds. *To design a substantially better decision procedure we should apply so-called top-down knowledge sources*, for example musical rules that determine the notes that are likely to play together or in sequence. Top-down processing will be discussed in Chapter 8. In the current system it is not used, but all note combinations and sequences are assumed equally probable.

7.6 Simulations

All the simulations were made using one instrument, piano. A percussive instrument was chosen to make rhythm tracking easier, while we do not want to focus the efforts on that. On the other hand, a piano sound is irregular enough to be a ‘real world’ example:

1. Beginning transient contains noisy components. Further, the sinusoid partials are in slightly inharmonic relations during the beginning tens of milliseconds, due to the strike of a hammer at strings.
2. Shape of the body of a piano produces several *formants*: frequency bands where harmonic partials are louder and decay slower because of the resonance of the body. The different decay slopes also cause the sound colour to change gradually in the course of its playing.
3. The higher keys have several strings for each note to produce a loud enough sound. Because the strings cannot be exactly equally tuned, they cause long term fluctuation in the amplitude envelope of the tone.
4. *Cross-resonance*. Playing just one note on the keyboard makes also the other free strings to start gently resonate along with the hammered string.

Two pianos were used. Initial simulations were made by sampling an electric piano in a noise-free environment. In the final simulations we used an acoustic upright piano (Yamaha) and the recording setup that was described in Section 4.3.

Three types of recordings were made. First, the keys were recorded one-by-one to provide a training set to the tone model creation process. Second, difficult note mixtures of several kinds were recorded to review the performance of the algorithm in possible stumbling points. Third, chosen musical pieces were recorded to test practical transcription efficiency.

All recordings were made only once and after that the recording setup was dismantled. This is both an advantage and a drawback. On one hand it makes the simulations reliable, while test material cannot be further adjusted to yield good results. On the other hand, this does not allow later reviewing of points where we suspect that the pianist may have unintentionally played some notes extremely quietly when recording rich note mixtures.

It should be noted that in all simulation cases, the transcription is done without knowledge of the polyphony of the transcribed signal or the range of notes that are involved in it. In practice, the range of notes was restricted to extend from the second octave *c* (65 Hz) up to the seventh octave *c* (2093 Hz), where 61 piano keys fit in between. Polyphony was limited to 30 simultaneous notes, which is by far enough for the chosen pieces. Figure 21 depicts the piano keyboard and indicates the included range of notes.

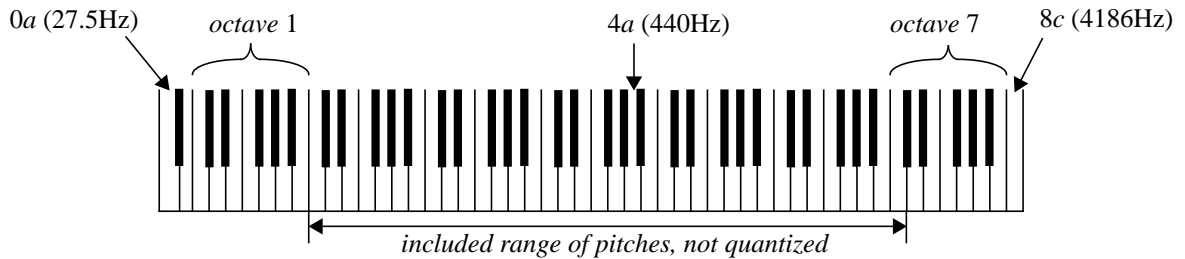


Figure 21. Compass of the piano keyboard, included range of pitches indicated.

Rhythm tracking

The rhythm tracking scheme was presented in Chapter 4, where simulation results for both polyphonic piano music and rhythm music recordings were given. Here the chosen instrument, piano, has a crucial role: rhythm tracking would not have been this successful with a non-percussive instrument, for example an organ.

Difficult note mixtures: review of resolving efficiency

Certain note mixtures were separately played and recorded to later review the ability of the algorithm to resolve polyphonic mixtures that are especially difficult from some point of view, and typically pose a stumbling point to transcription systems. Indeed, most transcription systems have not been able to deal with these cases. We fed these recordings to our transcriber, and the same figures of merit are monitored for each.

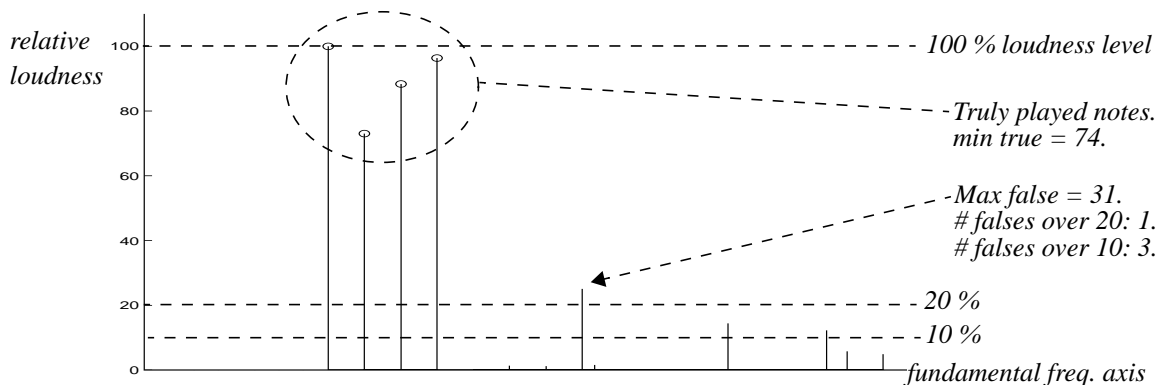


Figure 22. Interpretation of the results of an example simulation case (row 4 in Table 10).

In Figure 22 we illustrate how the simulation results should be interpreted. Naturally, there are much more potential note candidates than truly played notes. We call *true notes* the notes that were truly played into the recorded signals, and *false notes* those that appear as note candidates, although they were not actually played. To evaluate the efficiency of the transcriber in giving prominence to the true notes, we represent the results before the final segregation which

will throw away false notes based on their poor loudnesses. Thus the loudness of the true note candidates should raise clearly above the loudness of the false ones. Loudnesses are represented in relative terms, so that the loudness of the loudest note is always 100 %. It would be desirable to keep loudnesses of the true notes above 40 % and that of false notes below 20 %. Thus we present the minimum observed loudness among the detected true notes and the maximum loudness of a false note. Further, we present the number of false notes rising over 20 % and 10 % loudness levels.

Another goal is to *find the exact fundamental frequencies of the notes*, while they are *not* quantized to closest legal notes. This succeeded in all cases: relative errors are approximately 1%, while consecutive notes are separated by 6% frequency interval in piano keyboard.

Case 1. Major chords (notes *c, e* and *g* for *C major*) and *decremented plus sixth chords* (notes *c, d#, f#* and *a* in *C^oadd6*) were played at several positions of the keyboard and transcribed. These note combinations are quite common, but not trivial to transcribe, as stated earlier. The results are presented in Table 10. An acoustic piano gave rise to a certain phenomena that was not met with electric piano recordings: false notes highlighted with an asterisk (*) have quite high prominence with no evident reason. We interpreted this to be due to the cross resonance of the played notes, which strengthens certain harmonics and causes the tones to slightly differ from the tone models, which were built playing the notes alone.

Table 10: Chords played at different positions of the keyboard (case 1).

Difficulty: nothing special, fundamental frequencies in rational number relations.

Octave: chord	Max error in f_0 value (%).	Relative loudness		Undetected true notes	False notes above	
		min true	max false		20	10
2: <i>D major</i>	0.3	90	21	-	1	5
2: <i>C^oadd6</i>	0.3	62	26*	-	1	3
3: <i>E major</i>	0.6	78	43*	-	1	1
3: <i>C^oadd6</i>	0.5	74	31*	-	1	3
4: <i>F major</i>	1.0	49	4	-	-	-
4: <i>C^oadd6</i>	0.9	83	12	-	-	1
5: <i>G major</i>	1.3	57	0	-	-	-
5: <i>C^oadd6</i>	1.1	41	6	-	-	-
6: <i>A major</i>	2.2	38	8	-	-	-
6: <i>C^oadd6</i>	0.6	66	16	-	-	-

Case 2. Playing together two notes, whose fundamental frequencies are in

$$f_0_2 = m \cdot f_0_1$$

relations. In this case the lower note totally overlaps the higher one, and we can review the efficiency of the subtraction principle. The root note is *d* in the third octave of a piano and the second note is played at the positions of the harmonics of the root note. The results are presented in Table 11. For the sake of convenience we wrote the number of corresponding harmonic in parenthesis after the notes. It turns out that the subtraction principle reveals and clearly detects also the higher note.

Case 3. C^oadd6+octave chord (seven notes, *c, d, e, g, a, a#* and second *c*) were played at several positions of the keyboard and transcribed. The results are given in Table 12. We must notify two things. First, the subtraction principle no longer revealed the higher *c* in this polyph-

Table 11: Notes in straight harmonic relations (case 2).

Difficulty: lower note totally overlaps the harmonics of the higher one.

Octave: notes (No of harm.)	Max error in f_0 value (%).	Relative loudness		Undetected true notes	False notes above	
		min true	max false		20	10
3:d 4:d (2)	0.6	69	14	-	-	1
3:d 4:a (3)	0.5	41	15	-	-	1
3:d 5:d (4)	1.5	76	10	-	-	1
3:d 5:f# (5)	1.2	92	18	-	-	1
3:d 5:a (6)	1.1	63	6	-	-	-
3:d 6:c (7)	1.5	68	8	-	-	-
3:d 6:d (8)	1.3	90	8	-	-	-

ony, because its harmonics can too easily be explained to be derived from the other notes, especially the lower *c*. It was undetected in all cases (see *s), except two, where it was detected, but estimated very quiet. The positive notification is more important, however: all the other notes were detected even in the low octaves, and no false notes raise from the mass of candidates. The quite low loudness of the minimum true note should not be wondered, since it is partly due to the fact that a pianist does not play notes with equal loudnesses in this rich polyphony.

Table 12: Seven notes played at several positions of keyboard (case 3).

Difficulty: rich polyphony, many fundamental frequencies in rational number relations.

Octave	Max error in f_0 value (%).	Relative loudness		Undetected true notes	False notes above	
		min true	max false		20	10
2	0.6	25	11	1*	-	3
3	0.6	43	6	1*	-	-
4	0.9	58	10	1*	-	1
5	1.4	48	4	1*	-	-
6	2.2	18	0	1*	-	-

Case 4. Random mixture of seven notes. For the sake of convenience this single test was performed on an electric piano. Seven random notes were played together, using the decimals of pi as a random number generator ($\pi=3.14159265\dots$). We numbered the notes of the keyboard, choosing the note *c* in third octave to be note zero. The first set of seven notes is formed as follows: the first note is number 3 (*d*), and the others are $3+\underline{1}=4$ (*d#*), $4+\underline{4}=8$ (*g*), $8+\underline{1}=9$ (*g#*), $9+\underline{5}=14$ (*c#* in the next octave), and so on. The next set of notes is formed in the same manner using the next seven decimals of pi. In the last two sets decimal zero was encountered, which meant that the same note was played twice, i.e., two times as loud as the others. This partly explains the low loudness highlighted with * (see Table 12), because it is relative to the loudest one.

Case 5: Root note plus chords. The fundamental frequencies of the notes constituting a major chord (*c*, *e* and *g* for *C major*) correspond to the fourth, fifth and sixth harmonics of the *root note* of the chord, which is two octaves below the chord. Because of its nice harmonic properties, the root note is often played together with the chord, and in this case the root note overlaps all the higher notes. In Table 14 we present results from cases where chords are played at different octaves with their root note.

Table 13: Random mixture of seven notes played together on an *electric piano* (case 4).
Difficulty: rich polyphony.

Octave note, octave note, ...	Max error in f_0 value (%).	Relative loudness		Undetected true notes	False notes above	
		min true	max false		20	10
<i>3d, 3d#, 3g, 3g#, 4c#, 4a#, 5c</i>	0.3	25	17	-	-	2
<i>3f, 3a#, 3c#, 3f#, 4d, 4h, 5f#</i>	0.5	25	10	-	-	1
<i>3g#, 3h, 4c#, 4e, 5c, 5d#, 5a</i>	0.3	23	8	-	-	-
<i>3c#, 3g, 3h, 4d, 4f, 5c#, 5e</i>	0.2	10	11	1	-	1
<i>3c#, 3g#, 4f, 4a#, 4a#, 5c, 5g#</i>	0.2	6*	3	-	-	-
<i>3g, 3h, 4c, 4a, 5e, 5g, 5g</i>	0.3	36	16	-	-	2

A certain problem, which is again highlighted with *s, was detected in our system. There is no true note at the positions of the second and third harmonics of the root note. However, if the subtraction of the root note leaves a fundamental candidate to these positions, that false note may partially ‘steal’ the energies of the higher notes, and may appear even as a very loud note, as can be read from the table. In the first occurrence a true note moved one octave downwards because of a false note at the third harmonic thieving the energy of a true note at sixth. In the second occurrence no true notes remain undetected, but a gross false note arises over the trues. This kind of gross errors are not allowed, and thus they led to a new scheduling in processing the candidates. The new scheme is not presented here, since it requires still more experimenting.

Let us still have a look at the two last rows of the table. There the base note is still in a harmonic relation to the notes of the chord, but is not the root note. In that case the abovementioned problem did not appear.

Table 14: Chords with a root note (case 5).

Difficulty: notes of the chord are totally overlapped by the root note.

Octave: notes	Max error in f_0 value (%).	Relative loudness		Undetected true notes	False notes above	
		min true	max false		20	10
<i>2:c 4:Cmajor</i>	0.5	30	30	1*	2	2
<i>2:c 4:Cminor</i>	0.9	23	18	-	-	2
<i>3:d 5:Dmajor</i>	1.1	59	8	-	-	-
<i>3:d 5:Dminor</i>	0.9	56	42	-	1	1
<i>4:e 6:Emajor</i>	1.5	42	13	-	-	1
<i>4:e 6:Eminor</i>	1.6	27	250*	-	1	1
<i>2:g 5:Cmajor</i>	1.3	39	6	-	-	-
<i>3:e 5:Cmajor</i>	1.2	39	10	-	-	-

Case 6. Several adjacent notes. These notes are close to each other in pitch, but are not in a harmonic relation to each other, and were easily detected. The true note is missed only once. The problem of a weak note candidate being masked by a close lying stronger note was solved by the way the f_0 candidates were picked from the spectrum - see page 53 to recall what it means

to *widen* a spectrum peak.

Table 15: Several notes close to each other in pitch (case 6).

Difficulty: a strong tone easily masks a close relying weaker one.

Octave: notes	Max error in f_0 value (%)	Relative loudness		Undetected true notes	False notes above	
		min true	max false		20	10
2: <i>e f f# g g#</i>	0.6	39	5	-	-	-
3: <i>f f# g g# a a#</i>	0.9	40	2	-	-	-
3: <i>e f _ g g#</i>	0.5	87	17	-	-	2
4: <i>g g# _ a# h</i>	0.9	40	7	-	-	-
5: <i>a a# h 6:c c#</i>	1.8	30	7	-	-	-
5: <i>f f# _ g g#</i>	1.1	49	3	1	-	-

Transcribing polyphonic segments of chosen compositions

Chosen compositions were recorded and excerpts from them were fed to our transcription system. In principle this is the very same as resolving the note mixtures in the previous sections, provided that we first have a rhythm tracking procedure to split the signal at note onset points. An additional factor here is that some notes may continue ringing over several signal segments and their tone slightly changes through time, since the formants of the instrument make some harmonics cease slower than the others, making the tone models only approximate the true tone colour. Rhythm imperfections and loudness variations among concurrent sounds also bring new phenomena.

We present the results after applying a 25 % loudness limit to segregate between true and false notes. This allows us to present the statistics more compactly.

Composition 1: The beginning twelve measures of Bach’s “Inventio 8” were played and fed to our transcriber. The results are shown in Table 16. As expected, in this piece of just two-voice polyphony almost all notes were correctly transcribed. All undetected true notes (see *) were missing just because of a rhythm tracking error. Further, seven of the nine true notes below 25% (see **) were in a certain sequence, where two notes alternate and the other one is systematically played so quietly that it falls below the limit. The draft notation produced by the transcriber would have worked well as a helping tool for a musician.

Table 16: Inventio 8, beginning twelve measures.

Composition	Notes in total	Typical polyphony	Lacking true notes		False notes above 25%
			Undetected	Below 25%	
Inventio 8	205	2	6*	9**	4

Composition 2: The beginning twelve measures of Beethoven’s “Für Elise”, repetitions omitted. This music extract is monophonic most of the time, but is not trivial, since it contains low

notes of only short time duration. The results are presented in Table 16.

Table 17: Für Elise, beginning twelve measures.

Composition	Notes in total	Typical polyphony.	Lacking true notes		False notes above 25 %
			Undetected	Below 25%	
Für Elise (I)	86	1	-	-	3

Composition 3. Another twelve measures extract from Für Elise, containing rapid note sequences and several four voice polyphonic periods. In the sections containing five or six note polyphonies only one true note was lacking. The results are presented in Table 16.

Table 18: Für Elise, another twelve measures.

Composition	Notes in total	Typical polyphony.	Lacking true notes		False notes above 25 %
			Undetected	Below 25%	
Für Elise (II)	190	half:2, half:4	4	5	6

Composition 4. The beginning twelve measures of Mozart’s “Rondo alla Turca”. This single piece was played by a computer on an electric piano. The effect of all notes having roughly equal playing loudness and the absence of cross resonance and noise can be noticed in the results of Table 16.

Table 19: Rondo alla Turca, beginning twelve measures.

Composition	Notes in total	Typical polyphony.	Lacking true notes		False notes above 25%
			Undetected	Below 25%	
Rondo alla Turca	142	3 (up to 5)	-	1	-

7.7 Comparison with a *reference system* utilizing straightforward pattern recognition

Music transcription is a complicated task and it is therefore difficult to see what advantages or errors are due to the new number theoretical method and what result from other properties of the whole system. However, according to our observations a certain good property was attained just because of the new method, and it is *robustness in rich polyphonies*. To illustrate this we shortly review the results of an older reference system which was based on straightforward pattern recognition.

Before the presented algorithm was recovered, we developed a system that was aimed to solve the same problem as the one presented earlier: all the tones of a piano were first recorded to form a palette of possible sounds, and the program was then asked to resolve a mixture of several sounds into a limited set of those known tones. In some aspects that older system is more sophisticated than the one that was presented above. It utilizes the constant Q transform to map an optimally precise time-frequency spectrum of the tones [Brown88,92]. These two dimensional tone models took into account the time evolution of the frequency contents, which was ignored in the system presented in this chapter. Tone models looked like the one presented in Figure 23.

It turned out, however, that tone patterns in the straightforward pattern recognition approach

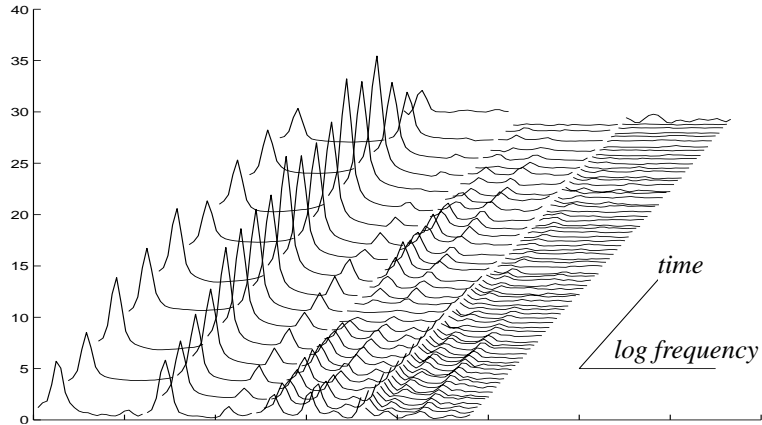
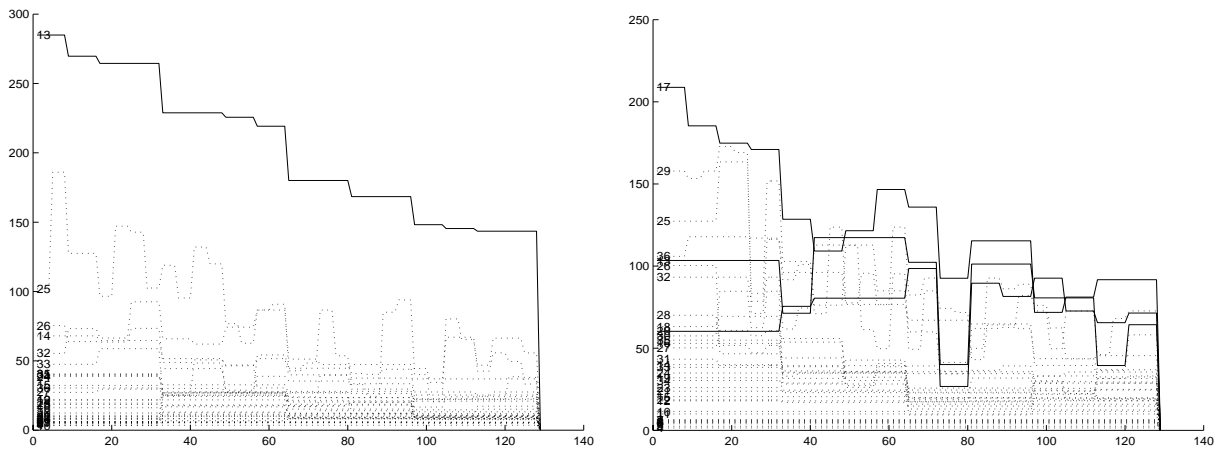


Figure 23. Tone model in the reference transcription system (piano note *d* in octave 4).

are completely sunk under each other already in three notes polyphony. This is due to the properties of an overlapping harmonic series and the percentages of the overlapped amounts that were discussed in the previous chapter. Figure 24 illustrates transcription results of that reference pattern recognition approach. In that figure we plot the degree of match of each tone model as a function of time. Truly played notes are plotted with solid lines, false candidates with dashed lines. The higher is the line, the better the tone matches the signal. The decreasing tendency is due to the ceasing of the percussive piano sound. It can be seen that a monophonic signal is still quite well transcribed (figure 24a), but in a major chord of three notes *c*, *e* and *g*, the true notes are already totally immersed among the false ones (figure 24b).



a) Monophonic signal (note *c* in third octave).
The true note raises clearly over the false ones.

b) Three notes polyphony (*C major* in third octave). The true notes sink among the false ones.

Figure 24. Reference transcription system: degrees of match of the tone models through time.

7.8 Conclusion

The problem of transcribing rich polyphonies was the motivation for developing the new methods. Simulations illustrate that the current system works within certain error limits up to seven notes polyphony. Especially, although increase in polyphony brings the levels of the weakest true note and the strongest false note closer to each other, the system does not totally break down even in rich polyphonies. We conclude that a number theoretical analysis of a sound mixture is the key to detecting and observing the sounds robustly in the interference of each other.

7.9 Future additions

The new method was simulated only in a limited domain of musical signals and using a very limited set of psychoacoustic cues (listed on page 28). As extensively described in Chapter 6, the presented number theoretical method would suit the extraction of the other cues as well, not only for observing a momentary loudness of the harmonic partials. Utilizing additional information is likely to improve the results, but we want to be careful in saying this - integrating the different pieces of evidence is an area of only a little psychoacoustic knowledge.

A substantial improvement to the system would be to discard a separate training set. If our transcription system would just be able to extract its tone models from the musical piece itself, it would already be an efficient and general purpose transcription tool. Automatic, or adaptive tone modeling will be discussed in Sections 8.3 and 8.4. We experimented automatic tone modeling with a gross parametrization of tone colours by a second order polynomial and got promising results.

As mentioned earlier, tone models in our current system are a gross simplification of reality, since they do not take into account the time evolution nor the beginning transient of the tones. Time evolving models are important to recover tones that play over several rhythmic segments of the signal and change in the course of their playing. The beginning transient is of fundamental importance in distinguishing different instruments from each other [Meillier91], but also in resolving rich polyphonic contents where several notes are in straight harmonics relations and two instruments may even play a same note in unison. The beginning transient also often reveals the notes that are intentionally played very quietly. Instrument models will be thoroughly discussed in a coming Section 8.4.

An algorithmic improvement to our system could be made by utilizing iterativeness. Our algorithm is not likely to totally break down with less than ten notes polyphony, but the error in estimated loudnesses steadily increases along with an increasing amount of polyphony, because more harmonics of the sounds are being overlapped by each other. This could be solved by first calculating the loudnesses of note candidates as before, second, choosing the candidates that are most promising to be the true notes, and third, recalculating the loudnesses using only the harmonics that cannot be corrupted by the most promising candidates.

Finally, a significant portion of the errors in the output of the current system could be removed by applying even primitive top-down reasoning. These processing mechanisms will be represented in the next chapter.

8 Top-Down Processing

Until now, our discussion on music transcription has almost exclusively concerned bottom-up processing techniques. They are characterized by the fact that all information flows bottom-up: information is observed in an acoustic waveform, combined to provide meaningful auditory cues, and passed to higher level processes for further interpretation. This approach is also called *data-driven* processing.

Top-down processing utilizes internal, high-level models of the acoustic environment and prior knowledge of the properties and dependencies of the objects in it. In this approach information also flows top-down: a sensing system collects evidence that would either justify or cause a change in an internal world model and in the state of the objects in it. This approach is double-called *prediction-driven* processing, because it is strongly dependent on the predictions of an abstracted internal model, and on prior knowledge of the sound sources [Ellis96].

The science of perception and the perceptual models of audition are dominated by a bottom-up approach, most often ignoring top-down flow of information in the human auditory system [Slaney95]. In our transcription system, the only top-down mechanism is its use of tone models, i.e., prior knowledge of the instrument sounds. On the other hand, musical knowledge was totally ignored, assuming all note combinations and sequences equally probable.

The foundation of signal analysis is still in reliable low-level observations. Without being able to reliably extract information at the lowest level, no amount of higher level processing is going to resolve a musical signal. It is therefore clear that top-down processing cannot replace the functions of a bottom-up analysis. Instead, *top-down techniques can add to bottom-up processing and help it to solve otherwise ambiguous situations*. Top-down rules may confirm an interpretation or cancel out some others. On the other hand, high-level knowledge can *guide* the attention and sensitivity of the low-level analysis.

It should be noted that top-down processing does not require that the information should have *originated* at a high-level process. Conversely, all top-down processing should be highly *interactive* and adapt to the signal at hands. For example, context is utilized when information is collected at a low level, interpreted at a higher level, and brought back to affect low-level processes. Similarly, internal tone models should originate from low-level observations.

A hierarchy of information representations, as discussed in Chapter 3, does not need to imply bottom-up processing: representations may be fixed, and still information flows both ways. Data representations, processing algorithms, and an implementation architecture should be discussed separately, according to the desirable properties of each.

In this chapter we first discuss the shortcomings of pure bottom-up systems and represent some top-down phenomena in human hearing. Second, we represent top-down knowledge sources that can be used in music transcription: use of context, instrument models, and primitive ‘linguistic’ dependencies of note sequences and combinations in music. Then we investigate how that information could be practically used, and finally propose some implementation architectures.

8.1 Shortcomings of pure bottom-up systems

The critique of pure bottom-up models of the human auditory system is not only due to the fact that they fail to model some important processes in human perception. The very basic reason for criticism is that often these models are not sufficient to provide a reliable enough interpretation of acoustic information that comes from an auditory scene. This applies especially to a general purpose computational auditory scene analysis, where some top-down predictions of an internal world model are certainly needed. For example, even a human listener will find it difficult to analyze the number of persons and their actions in a room by acoustic information only, without prior knowledge of the actual physical setting in that room.

Transcription of musical signals is a substantially more specialized task, and thus more tractable for bottom-up systems. Nevertheless, *inflexibility*, when it comes to the generality of sounds and musical styles, is a problem that is largely derived from systems' inability to make overall observations of the data and let that knowledge flow back top-down for the purpose of adapting the low-level signal analysis. Another lacking facet of flexibility is the ability to work with obscured or partly corrupted data, which would still be easily handled by the human auditory system.

In the following sections, we will represent knowledge sources and mechanisms that add the intelligence of bottom-up systems. That knowledge can be used to prevent the bottom-up analysis being misled in some situations, or to solve otherwise ambiguous mixtures of sounds.

8.2 Top-down phenomena in hearing

Psychoacoustic experiments have revealed several phenomena that suggest top-down processing to take place in human auditory perception. Although this has been known already for years, explicit critique of pure bottom-up perception models has emerged only very recently [Slaney95, Ellis96, Scheirer96]. In this section we review some phenomena that motivate a top-down viewpoint to perception.

Auditory restoration refers to an unconscious mechanism in hearing, which compensates the effect of masking sounds [Bregman90]. In an early experiment, listeners were played a speech recording in which a certain syllable had been deleted and replaced by a noise burst [Warren70]. Because of the linguistic context, the listeners also 'heard' the removed syllable, and were even unable to identify exactly where the masking noise burst had occurred.

Slaney describes an experiment of Remez and Rubin which indicates that top-down processing takes place in organizing simultaneous spectral features [Slaney95]. Sine-wave speech, in which the acoustic signal was modelled by a small number of sinusoid waves, was played to a group of listeners. Most listeners first recognized that signal as a series of tones, chirps, and blips with no apparent linguistic meaning. But after some period of time, all listeners unmistakably heard the words and had difficulties in separating the tones and blips. The *linguistic information* changed the perception of the signal. In music, internal models of the instrument sounds and tonal context have an analogous effect.

Scheirer mentions Thomassen's observation, which indicates that *high-level melodic understanding* in music may affect the low-level perception of the attributes of a single sound in a stream [Scheirer96a]. Thomassen observed that certain frequency contours of melodic lines lead to a percept of an *accented* sound - as it would have been played stronger, although there was no change in the loudness of the sounds [Thomassen82].

Slaney illustrates the *effect of context* by explaining Ladefoged's experiment, where the same constant sample was played after two different introductory sentences [Slaney95]. Depending on the speaker of the introductory sentence "Please say what this word is: -", the listeners heard the subsequent constant sample to be either "bit" or "bet" [Ladefoged89].

Memory and hearing interact. In Chapter 6.3 we stated that paying attention to time intervals in rhythm and to frequency intervals of concurrent sounds has a certain goal among others: to unify the sounds to form a coherent structure that is able to express more than any of the sounds alone. We propose that also the *structure* in music has this function: similarities in two sound sequences tie these bigger entities together, although they may be separated in time and may differ from each other in details. These redundancies and repetitions facilitate the task of a human listener, and raise expectations in his mind. Only portions of a common theme need to be explicitly repeated to reconstruct the whole sequence in a listener's mind, and special attention can be paid to intentional variations in repeated sequences.

8.3 Utilization of context

Certain aspects of the human auditory restoration can be quite reliably modelled, if the low-level conditions and functions of that phenomenon can just be determined. These include *illusory continuity* either in time or in frequency dimension. If a stable sound becomes broken for a short time and an explaining event (such as a noise burst) is present, that period can be interpolated, since the human auditory system will do the same. On the other hand, if the harmonic partials of a sound are missing or corrupted at a certain frequency band, the sound can still be recovered by finding a harmonic series at adjacent bands.

The auditory restoration example in Section 8.2 represents a type of restoration that is based on linguistic knowledge and cannot be done as easily. That requires applying higher level knowledge. Primitive 'linguistic' dependences in music will be discussed in Sections 8.5 and 8.6.

In Section 5.4 we made a point that separating harmonically related sounds is theoretically ambiguous without internal models of the component sounds. These models do not necessarily need to be pre-defined, but may be found in the context of an ambiguous mixture. Internal models should be assembled and adapted at moments when that information is laid bare from under the interference of other sounds, and used at more complicated points. It is well known that if two sound sequences move in parallel in music, playing the same notes or two similar melodies with a harmonically related frequency separation, the auditory system will inevitably consider them a single unit. Typically this is not the composer's intention, but music must introduce the *atomic patterns*, sounds, by representing them as varying combinations. By recovering these atomic sounds, melodic lines of two instruments can be perceptually separated from each other, although they would instantaneously totally overlap each other.

Musical signals are redundant at many levels. Not only do they consist of a limited set of instrumental sounds, but they also have *structural redundancy*. The ability of a listener to follow rich polyphonies can be improved by a thematic repetition, and by adding new melodic layers or variations one by one. This strategy is often employed, for example, in classical and in jazz music, when the richness and originality of climax points could not be directly accepted by a listener.

8.4 Instrument models

In our transcription system, tone models were the only top-down knowledge source. However, careful compilation of these models, coupled with the presented number theoretical methods could solve transcription problems that have been earlier impossible. Indeed, we want to emphasize that the use of instrument models is a powerful top-down processing mechanism, provided that this information can be collected. Even more, we showed in Section 5.5 that these models are indispensable because of the theoretical ambiguity that will otherwise be met in separating harmonic mixtures of musical sounds. Separating them without instrument models is possible only if there are other distinguishing perceptual cues. However, practically in every single piece of music there will be a multitude of note mixtures that cannot be resolved without instrument models.

This does not mean that instrument models must be built beforehand from a training material. They can be assembled either from separate training material, or during the transcription from the musical signal itself, or by combining these two: starting from a set of standard instrument models that are then *adapted*, reconciled using the musical signal. We use the terms *tone* model and *timbre* model in different meanings, the former referring to a model of a sound at a certain pitch, and the latter to the model of instrument characteristics that apply to its whole range of sounds. The term timbre alone means the distinctive character of the sounds of a musical instrument, apart from their pitch and intensity, ‘sound colour’. It seems that the human auditory system uses timbre models, since it tends to model the source, not single sounds.

If our transcription system were just able to extract its tone models from the musical piece itself, it would be an efficient and general purpose transcription tool. A critical problem in doing that is to *recognize* an instrument, to know what model to use when needed, or to reconcile by adapting when it is possible. Controlled adapting by accounting only the non-interfered partials would then be possible, based on the analysis in Chapter 6. Another problem then is to find a parametrization for the models that allows using the information of just one sound in refining the timbre model of the whole instrument.

Since we think that distinguishing different instruments from each other is of critical importance, we concentrate on studying *what is the information that should be stored in the models*, i.e., what are the types of information that give a distinctive character to the sounds of an instrument. Parametrization and use of that information goes beyond the scope of this thesis.

We have taken a *dimensional approach* in organizing and distinguishing timbres of different instruments. This means that their relations or similarities cannot be expressed using just one attribute, but a timbre can resemble another in different ways, as well as a physical object may resemble another in size, shape, material, or colour. A classic study on organizing timbres to a feature space of features was published by Grey in 1977 [Grey77]. Grey found three distinctive dimensions of timbres. 1) The brightness of sounds: spectral energy distribution to high and low harmonic partials. 2) The amount of spectral fluctuation through time, i.e., the degree of synchronicity in the amplitude envelopes of the partials. 3) Presence and strength of high frequency energy during the attack period of the sounds.

We propose a selection of dimensions that pays attention to the fact that a human tends to hear the *source* of a sound, and understands a sound by imagining its source. Therefore we propose that the fundamental dimensions of a timbre space are properties of sound sources, factors that leave an informative ‘fingerprint’ to the sound, and enable a listener to distinguish different

sources, instruments, from each other. These dimensions bear resemblance to those of Grey, but the attention is not paid to the signal but to its source, which we consider more natural, informative, and descriptive. The ‘fingerprints’ to a sound are set by

1. Properties of the vibrating source.
2. Resonance properties of the body of a musical instrument. That is the immediate pathway of the produced sound, intensifying or attenuating certain frequencies.
3. Properties of the driving force that sets a sound playing. The mechanism how it interacts with the vibrating source to give birth to a sound.

It is a well known fact that the human auditory system is able to separate the first two aspects above [Bregman90]. Since especially the third aspect is usually ignored in the literature, we will discuss it more than the others. It seems that these three properties of the source effectively explain the information that is present in a resulting acoustic signal.

Vibrating source

A vibrating source is the very place where a sound is produced. It causes at least two kinds of spectral features. First, brightness of a sound, which is determined by the energy distribution between high and low frequency components. Brightness was found to be the most prominent distinguishing feature in the experiments of Grey. Second, a vibrating source often produces certain regularities in the harmonic series of a sound. In a clarinet sound, for example, odd harmonics are stronger than the even ones.

Body of a musical instrument

The term *formant* refers to any of the several characteristic bands of resonance, intensified frequency bands that together determine the quality of a spoken vowel, for example. The structure of the body of a musical instrument causes formants. Its size, shape and material altogether make it function as a filter, which intensifies some frequency bands and attenuates some others. Harmonic partials at formant frequencies play louder and decay slower, which causes the timbre of the sound to smoothly change in the course of its playing.

One good way to model the body of an instrument is to determine its frequency response. The pattern of intensified and attenuated frequency bands affects the perceived timbre of a sound a lot. Bregman clarifies this by taking different spoken vowels as an example. Their distinction depends on the locations of the lowest three formants of the vocal tract. The lowest two are the most important and are enough to provide a clear distinction between the vowels [Bregman90].

Driving force

A musical sound typically consists of an attack transient followed by a steady sound. This structure derives itself from the physical production of a sound: the driving phenomenon that sets the sound playing leaves its ‘fingerprint’ to the sound before it stabilizes or starts decaying. A human listener is able to recognize, for example, the way in which a guitar string is plucked, or to hear out the noisy turbulence of air in the beginning of the sound of a transverse flute. The transient, in spite of its very short duration, is very important for the perceived quality of the sound [Meillier91]. An analysis of musical instrument sounds shows that a large portion of the spectral information of the sound comes out during the first tens of milliseconds. This applies especially to percussive sounds [Meillier91]. The beginning information burst might be utilized as a ‘label’ introducing the sound source.

Measuring the attack transient

Frequency transforms that were discussed in Chapter 3, such as FFT or bounded Q transform, are not very suitable for measuring the precise time evolution of frequency components, since they are designed to be effective in providing frequency information in one time frame, not in observing the time evolution of certain frequencies, when extremely short time frames and analysis steps must be used. The usefulness of several high-resolution spectral analysis methods were evaluated in order to yield precise information of the attack transient [Meillier91, Laroche93, Kay88]. A crucial drawback of the methods in our simulations was that they are computationally complex and also sensitive to the choice of analysis parameters and to the signal to which they are applied. This prevented us from their practical utilization. In the following, we propose a quite straightforward alternative method for extracting precise time evolving amplitude envelopes of the frequency components of interest.

The Fourier transform $X(k)$ of a sampled time domain signal $x(nT)$ is calculated as

$$X(k) = \sum_{n=0}^{N-1} \left(x(nT) \cdot e^{-j \cdot k \cdot \frac{2\pi nT}{N}} \right), \quad (30)$$

where k is a discrete frequency bin, zero corresponding to zero frequency, and $k = \frac{N}{2}$ to half the sampling rate. This transform can be effectively calculated over the whole frequency band in certain time frame using the fast Fourier transform. We pay attention to the fact that it can also be efficiently calculated at successive time frames for a certain frequency bin.

When a certain frequency bin is under inspection, the length of the time domain analysis frame can be adjusted to be a multiple of the wavelength of that frequency. In this case the length of the time frame may be very short (say, three waves), the windowing function in time domain is not necessary, and we can extract a precise sample-by-sample amplitude envelope of that frequency through time. This is done by first calculating the transform for a certain frequency bin as usual, and storing all the elements of the sum in Equation 30. Now the transform of the same bin in a time frame one sample later can be calculated just by subtracting the first element of the sum, and adding a new element, which is calculated by

$$x(nT) \cdot e^{-j \cdot k \cdot \frac{2\pi nT}{N}}, \quad (31)$$

where n now points to the sample that is right after the previous frame.

A property of the sinusoid track representation of sound (see Chapter 3) is that it is compact and still bears high perceptual fidelity to the original sound. The peaks in the BQT spectrum of the beginning of a sound reveal the sinusoids of interest, and therefore also the frequency bins, whose amplitude envelope should be followed to model that attack of a sound. The proposed spectral analysis method is efficient, since the number of sinusoids is limited (typically 10-50), and the attack transient over which their amplitude envelope is tracked is very short (about 50 ms). A more advanced variant of this method could also follow the frequency fluctuations of a sinusoid, but representing its details is not relevant in our scope.

8.5 Sequential dependencies in music

Most speech recognition systems today use linguistic information in interpreting real signals where single phonemes may be obscured or too short to be recognized only by the information in a single time frame. Linguistic rules are typically very primitive, such as a table of statistical

probabilities for the existence of certain two or three letter sequences in a given language. This kind of data can be utilized in a *hidden Markov model (HMM)*, which basically implements a state machine, where single letters are the states, and probabilities for transitions between the states have been calculated in the language. Based on both low-level observations and these high-level constraints, a sensing system then determines the most probable letter sequence.

In [Scheirer96a], Scheirer asks whether primitive ‘linguistic’ dependencies of this kind could be found for note sequences in music, and under what circumstances following single notes and polyphonic lines is possible, difficult or impossible for humans. Fortunately, Bregman has studied these dependencies and has collected the results of both his own and other researchers’ work in [Bregman90]. Since that information has been practically ignored by the music transcription community, we will shortly review it in the following. Bregman’s intention is not to enumerate the compositional rules of any certain musical style, but rather to try to understand how the primitive principles of perception are being used in music to make complicated sound combination suit human perception.

Sequential coherence of melodies - a review on Bregman’s results

Both time and frequency separations affect the integrity of perceived sequences, according to Bregman’s experiments. These two have such a correlation that as the frequency separation becomes wider, a note sequence must slow down in order to maintain its coherence. The duration of a note in Western music typically falls in the decade between one and one tenth of a second, and if a note is shorter than this, it tends to stay close to its neighbors in frequency, and is used to create what Bregman calls an *ornamental* effect. An ornament note is perceived only in a relation to another, and does not itself help to define a melody.

Note sequences can be effectively integrated together by letting them advance by small pitch transitions. To illustrate that, Bregman refers to the results of Otto Ortmann, who surveyed the sizes of the frequency intervals between successive notes in several classical music compositions, totalling 23000 intervals [Ortmann26]. He found that the smallest transitions were the most frequently used, and the number of occurrences dropped roughly in inverse proportion to the size of the interval. What is most interesting, harmonic intervals do not play a special role in sequences, but it is only the size of the difference in log frequency that affects sequential integration. Sequential integration by frequency proximity can be illustrated by letting an instrument play a sequence of notes, where every second note is played at a high frequency range and every other at a low range. This will be inevitably perceived as two distinct melodies: frequency proximity overcomes the temporal order of notes. A melody may be unbroken over a large frequency leap only in the case it does not find a better continuation [Bregman90].

Timbre is an important perceptual cue in grouping sounds to their sources of production. This is why timbral similarities can be effectively used for the purpose of grouping sounds in music, and for carrying a melody through the sounds of an accompanying orchestration [Erickson75].

As mentioned earlier, Bregman does not intend to enumerate musical rules that go above those that can be explained by universal perceptual principles. However, he pays attention to a certain higher-level sequential principle that seems to be justified by some general rule of perception. That is the fact that a dissonant combination of notes is usually followed by a note or a mixture of notes which is more stable in the musical system and is able to provide a cognitive reference point, a ‘melodic anchor’. Furthermore, these inharmonious combinations of notes typically do not set on together with other notes, but are placed in between the strongest rhyth-

mic pulses to serve as a short moment of tension between two stabler states.

8.6 Simultaneity dependencies

In Section 6.3 we discussed how music uses harmonic frequency relations to group concurrent notes together. This takes place for two reasons. First, notes must be knitted together to be able to represent higher-level forms that cannot be expressed by single or unrelated atomic sounds. Second, the spectral components of harmonically related sounds match together, which effectively *reduces the complexity* of calculations that are needed in the human auditory system to group frequency components to acoustic entities. Harmonically related sounds appear merely as one coherent entity and are more easily followed.

Ironically, the perceptual intentions of music directly oppose those of its transcription. Dissonant sounds are easier to separate, and a computer would have no difficulty in paying attention to whatever number of *unrelated* melodies and note combinations, but has critical problems in resolving *chimeric* (fused) note combinations into their member notes. Human perception does not want to break down chimeras, but listens to them as a single object. Therefore music may recruit a large number of harmonically related sounds - that are hard to transcribe - without adding much complexity to a human listener. On the other hand, music has strict rules for the use of dissonances - that are easily detected by a transcriber - since they appear as separate objects for a listener.

In the previous section we already stated some regularities in the use of dissonance. Since transcribing dissonances is not the bottleneck of transcription systems, we consider it irrelevant to list more rules that govern it. Instead, we propose a principle that can be used in resolving rich harmonic polyphonies.

Dependencies in production

We spent time analyzing our personal strategies in listening classical and band music, and will here propose a top-down transcription procedure that seems to be natural, effective and musically relevant - in the sense that it is being used by musicians who need to break down chimeric sound mixtures for transcription purposes. The listening tests were mostly performed with band music called *fusion* (features from jazz, blues and rock), and with classical music.

The procedure that we suggest consists of two fundamental phases.

1. Recognize the type, and count the number of the instruments that are present in a musical performance.
2. Utilize two principles: *a)* Apply instrument specific rules concerning the range of notes that can be produced, and the restrictions on simultaneously possible sounds for each single instrument. *b)* Make global observations on the *grouping strategy* (described below) and repetitions of each instrument.

Sound production restrictions of an instrument can often be explicitly and unfailingly listed. For example, a saxophone can produce only one note at a time, has a certain range of pitches, and can vary in timbre only according to a known set of playing techniques. A guitar can produce up to six-voice polyphony, but the potential note combinations are significantly limited by the dimensions of the player's hand and potential playing techniques.

The auditory system naturally tends to assign a coherent sequence of notes to a common source, and larger structures of music are understood by interpreting them as movements,

transformations, or expressions of these sources of production [Bregman90]. This seems to explain what we have noticed, that distinct instruments tend to choose their own strategies of grouping notes to meaningful entities. A flute, for example, integrates notes by forming *sequences*. A guitar has more possibilities: in addition to the ability of producing sequences, it can accompany a piece of music by dropping groups of simultaneous notes, chords, that are integrated by harmonic frequency relations. They can be further integrated on a higher level by forming a certain rhythmic pattern over time. A *rhythmic pattern* may provide a framework for combining sequential and simultaneous grouping principles. For example, a piano may take a pattern of playing four unit sequences, where the first two are single notes, the third is a simultaneously grouped note combination, and the fourth is a single note again. The grouping strategy of an instrument may change in different parts of a piece, but typically remains consistent over several measures (see Section 4.4) in a musical notation. This enables predictive listening, which is important for a human listener.

Counting the number of instruments is not needed for a solo (one player) performance, where the restrictions on production can be readily used. The other extreme is classical symphony music, where counting the instruments will surely prove impossible in some points, but, interesting enough, also impossible to transcribe for even an experienced musician. Reasonable cases, such as telling the instruments in a classical chamber music, or standard pop music will not pose a problem for an experienced listener. Still resolving the instruments automatically is anything but a trivial task. The method of utilizing context (see Section 8.3) also seems promising for this purpose: instrumental sounds may be recovered because they take varying combinations with each other, and may sometimes even play monophonic parts in the piece.

Usage of *synthesizer instruments* introduces an interesting case, since it can produce arbitrary timbres and even arbitrary note combinations by preprogramming them to the memory of the instrument. However, we suggest that since music must be tailored to suit perception, analogous rules will apply to synthesized music as well. Usually certain entities called *tracks* are established, and each track practically takes the role of a single performer.

8.7 How to use the dependencies in transcription

“That style is actually not music at all, this is why it could not be transcribed.”

-Unacceptable Excuses, 1997, unpublished

We have represented psychoacoustic dependencies and restrictions that are being used in music to make it suit human perception. But so far we have given only a few cues, exactly where and how that knowledge can be used in an automatic transcription system. Actually we found only a very little literature on how to *apply* top-down knowledge in computational models. A single reference is the earlier mentioned work of Ellis [Ellis96].

Human perception is amazing in its *interactivity*: internal models, for example instrumental timbres, affect the interpretation of acoustic data, but at the same time the acoustic data create and reconcile the internal models. This is of critical importance: we must be careful not to reduce the generality and flexibility of a transcription system by sticking to predefined internal models, such as a detailed musical style or the peculiarities of a single instrument. Thus the first principle in using internal models is that they should be exposed to the acoustic context: the models should be adapted and reconciled at the points where the relevant information is laid bare, and then utilized at the more complicated or ambiguous points.

An important fact recovered earlier is that predictions guide the attention and affect the sensitivity of the auditory system. Therefore we know that even a quiet note in a position to form a transition between two other notes will inevitably be heard as intended, and such a weak note candidate can be surely confirmed by the top-down rules. On the other hand, surprising events must be indicated clearly in music to be perceived and not to frustrate inner predictions. Using this rule, a weak note candidate that is grouped neither sequentially nor simultaneously nor rhythmically can be canceled out, because it would most probably be interpreted as an accidental and meaningless artefact or inference by a human listener. An essential operation in automatic transcription is that of canceling out single erroneous interpretations, ‘false’ notes, and therefore we think that these rules would significantly improve its performance.

Top-down processing rules should not be used too blindly or confidentially, or they will start to produce errors. Using the rules for guiding attention and sensitivity is quite safe. On the contrary, assuming too much on the probabilities of different note sequences and combinations will certainly limit the generality of a system. We think that top-down rules should be primarily used in solving ambiguous or otherwise unsolvable situations, where a human listener also has to make a guess of the most probable interpretation. The musical style dependent excuse, which is quoted below this section’s title certainly does not satisfy a revolutionary musician.

In any case, there are universal perceptual rules that are not style dependent and can be used to reject certain interpretations and to favour others in an ambiguous situation. Musical styles vary, but all of them need to employ a strategy of some kind to integrate sounds into a coherent piece of music, which can also be followed by other listeners than the composer himself. Bregman takes an example from Bach’s keyboard music: as the polyphony of a composition increases, perceptual techniques in organizing the notes to meaningful groups increase, too.

8.8 Implementation architectures

The implementation architectures of different transcription systems are dominated by heuristic and quite inflexible approaches. Integrating new perceptual cues and processing mechanisms, or changing the relations of the old ones is laborious and difficult. Two latest transcription systems make an exception: that of the Osaka research group and that of Keith Martin - together with Ellis’s computational auditory scene analysis system. They are built upon a so-called *blackboard* architecture, which will now be shortly described.

Blackboard architecture

Blackboard architecture has been developed for the purpose of solving complicated, many-faceted reasoning problems that require integration of various kinds of knowledge [Carver92, Lesser95]. Sensing systems are an important application area. Blackboard architecture is advantageous because it allows relatively easy adding and removing of processing modules and a flexible way of defining the co-operation and competition between different knowledge sources.

The name blackboard comes from a physical arrangement where several experts are standing around a blackboard, working together to solve a problem. Each expert makes additions or changes according to his knowledge, and at the same time sees how the solution evolves as a result of the common effort.

The central component of the architecture is a *database of hypotheses*, the blackboard itself.

The hypotheses are linked together to either support or inhibit others, therefore the effect of new evidence propagates through the database. *Knowledge sources* are active, independent modules that may create and modify hypotheses. One of them may extract information from an acoustic signal and bring it to the database, while another may manipulate the data according to a reasoning that is based on top-down knowledge. The *control* of the system is typically defined in a separate scheduler, which estimates the usefulness of each knowledge source and evokes them accordingly.

Blackboard implementation in the transcription system of the Osaka group uses quite a large range of knowledge sources: perceptual rules (listed on page 28), instrument models, and statistical data concerning note and chord transitions in music [Kashino95]. Keith Martin's blackboard-based transcription system comprised only bottom-up processing modules at the time of its publication [Martin96a,b]. The computational auditory scene analysis system of Ellis is the most top-down oriented (prediction-driven) among these three [Ellis96].

An implementational architecture of course does not remove the need for the algorithm development that must take place in defining the control modules and their interaction. However, a flexible architecture, such as a blackboard, facilitates experimenting with different subsets of knowledge sources and making changes to their relations and priorities.

9 Conclusions

We have studied recent knowledge on the automatic transcription of music and related areas of interest. The purpose of this work has been to represent the state-of-the-art in music transcription, and to add something to it. What we consider equally important is to have represented the most significant viewpoints and clues that provoke further research and give directions to it. The computational transcription of polyphonic music is still too inflexible and limited in efficiency to be commercially applicable. However, the research is in a fruitful phase, with several promising areas of pursuit that just wait for earnest tries. It is those promising potentials that we want to highlight in this conclusion.

Sinusoid track representation was chosen at the mid-level. This is because it enables a separate treatment of the harmonic partials in subsequent processing, which we consider necessary in order to resolve musical polyphonies. It also allows explicit use of auditory cues in assigning the frequency components to their due sources of production.

None of the recent transcription systems utilizes the whole range of auditory cues that were listed in Section 5.5. Our system used only the two first cues. Using synchronous amplitude or frequency changes in organizing the harmonic partials would still add much, since a point of change in a frequency partial can be hidden by an overlapping partial only if it poses a temporally concurrent change. Also paying attention to spatial proximity seems to be a fertile area of pursuit. Psychoacoustic knowledge is very limited, when it comes to the integration of the different psychoacoustic evidence to yield an interpretation. Blackboard architecture seems to be useful in the experiments that are therefore needed.

Top-down processing is an abundant source of knowledge that can be used, not only in probabilistic reasoning, but also in the form of deterministic rules that exclude otherwise potential interpretations. Inspecting the simulation results of our system indicated that a significant portion of the errors could have been removed by applying even quite primitive top-down rules.

Presented number theoretical methods constitute the most important part of this thesis, since they provide the desperately needed reliability in observing sounds in polyphonic musical signals. All additional levels of logic, such as reasoning by auditory cues or matching internal models to the acoustic information rely on robust low-level observations. Hence we believe that the new methods play an important role for the whole chain of processing towards interpretations. A particular efficiency was shown in resolving rich polyphonies, which has been a fundamental point of weakness in previous transcription systems.

The most acute challenge for the design of transcription systems is automatic instrument modeling, i.e., finding the atomic sound sources, instruments, and extracting their attributes using only the polyphonic musical signal itself. This problem has been addressed at several instances of this thesis (see Sections 8.3 and 8.4), and solving it would remove the other fundamental weakness of transcriptions systems: inflexibility in regard to the used instrument sounds.

Ironically, this work does not end up only in a computer program, but in a research program. The time needed to release a substantially better and practically applicable transcriber depends on the amount of the commercial interest that this topic is able to attract.

References

- [Astola97] Astola, Kuosmanen. (1997). "Fundamentals of Nonlinear Digital Filtering". CRC Press LLC.
- [Bilmes93] Bilmes. (1993). "Timing is of the Essence: Perceptual and Computational Techniques for Representing, Learning, and Reproducing Expressive Timing in Percussive Rhythm". MSc thesis, MIT.
- [Bregman90] Bregman. (1990). "Auditory Scene Analysis". MIT Press.
- [Brown88] Brown. (1988). "Calculation of a Constant Q Spectral Transform". *J. Acoust. Soc. Am.*, Jan. 1991.
- [Brown89] Brown. (1989). "Calculation of a 'Narrowed' Autocorrelation Function". *J. Acoust. Soc. Am.*, Apr. 1989.
- [Brown91a] Brown. (1991). "Musical Frequency Tracking using the Methods of Conventional and 'Narrowed' Autocorrelation". *J. Acoust. Soc. Am.*, May 1991.
- [Brown91b] Brown. (1991). "Musical Fundamental Frequency Tracking using a Pattern Recognition Method". *J. Acoust. Soc. Am.*, Sep. 1992.
- [Brown92] Brown. (1992). "An Efficient Algorithm for the Calculation of a Constant Q Transform". *J. Acoust. Soc. Am.*, Nov. 1992.
- [Brown93] Brown. (1993). "Determination of the Meter of Musical Scores by Autocorrelation". *J. Acoust. Soc. Am.*, Oct. 1993.
- [Canright87] Canright. (1987). "A Tour Up the Harmonic Series". *1/1 - the Journal of the Just Intonation Network*, Vol. 3, No. 3, 1987.
- [Carver92] Carver, Lesser. (1992). "Blackboard Systems for Knowledge-Based Signal Understanding". Oppenheim and Nawab (eds.). "Symbolic and Knowledge-Based Signal Processing". Prentice Hall.
- [Casey93] Casey. (1993). "Distal Learning of Musical Instrument Control Parameters". Proceedings of the International Computer Music Conference, 1993.
- [Chafe82] Chafe, Mont-Reynaud, Rush. (1982). "Toward an Intelligent Editor of Digital Audio: Recognition of Musical Constructs". *Computer Music Journal*, Vol. 6, No. 1, Spring 1982.
- [Chafe85] Chafe, Jaffe, Kashima, Mont-Reynaud, Smith. (1985). "Source Separation and Note Identification in Polyphonic Music". CCRMA, Department of Music, Stanford University, California.
- [Chafe86] Chafe, Jaffe. (1986). "Techniques for Note Identification in Polyphonic Music". Proceedings of the International Conference on Acoustics Speech and Signal Processing, 1986.
- [Depalle93] Depalle, García, Rodet. (1993). "Tracking of Partial for Additive Sound Synthesis Using Hidden Markov Models". *IEEE Trans. on Acoustics, Speech and Signal Processing*, 1993.
- [Desain88] Desain, Honing. (1988). "The Quantization Problem: Traditional and Connectionist Approaches". AI and Music Workshop, St. Augustin, Germany, Sep. 1988.
- [Desain91] Desain, Honing. (1991). "Quantization of musical time: a connectionist approach". In Todd and Loy (eds.), "Music and Connectionism". Cambridge: MIT Press.
- [Desain92] Desain. (1992). "A (De)composable theory of rhythm perception". *Music Perception*, 9(4), 1992.
- [Desain95] Desain, Honing. (1995). "Music, Mind, Machine. Computational Modeling of Temporal Structure in Musical Knowledge and Music Cognition". Unpublished manuscript, Aug. 1995.
- [Ding97] Ding, Qian. (1997). "Processing of Musical Tones Using a Combined Quadratic Polynomial-Phase Sinusoid and Residual (QUASAR) Signal Model". *J. Audio Eng. Soc.*, Vol. 45, No. 7/8, 1997.
- [Doval91] Doval, Rodet. (1991). "Estimation of fundamental frequency of musical signals". Proceedings of the International Conference on Acoustics, Speech and Signal Processing, 1991.
- [Doval93] Doval, Rodet. (1993). "Fundamental frequency estimation and tracking using maximum likelihood harmonic matching and HMMs". *IEEE Trans on Audio Speech and Signal Processing*, 1993.
- [Ellis95] Ellis. (1995). "Mid-level representations for Computational Auditory Scene Analysis". Proceedings of the International Joint Conference on AI, Workshop on Computational Auditory Scene Analysis, Aug. 1995.
- [Ellis96] Ellis. (1996). "Prediction-driven computational auditory scene analysis". Ph.D. thesis, MIT.
- [Erickson75] Erickson. (1975). "Sound Structure in Music". University of California Press.
- [General91] MIDI Manufacturers Association and the Japan MIDI Standards Committee. (1991). "General MIDI System Level 1 specification". Sep. 1991.
- [Gerson78] Gerson, Goldstein. (1978). "Evidence for a General Template in Central Optimal Processing for Pitch of Complex Tones". *J. Acoust. Soc. Am.*, 63, 1978, pp. 498-510.
- [Grey77] Grey. (1977). "Multidimensional perceptual scaling of musical timbres". *J. Acoust. Soc. Am.*, Vol. 61,

No. 5, May 1977.

- [Grossberg76] Grossberg. (1976). "Adaptive Pattern Classification and Universal Recoding". *Biological Cybernetics*, 23, 1976, pp. 121-134.
- [Grusin91] Grusin. (1991). "Punta del Soul". On the record "Migration". GRP Records Inc.
- [Hawley93] Hawley. (1993). "Structure out of Sound". Ph.D. thesis, MIT.
- [Hess83] Hess. (1993). "Algorithms and Devices for Pitch Determination of Musical Sound Signals". Springer Verlag, Berlin.
- [Horn85] Horn, Johnson. (1985). "Matrix Analysis". Cambridge University Press.
- [Ifeachor93] Ifeachor, Jervis. (1993). "DSP-practical approach". Addison-Wesley Publishing Co.
- [ISO87] International Organization for Standardization. (1987). "Acoustics - Normal Equal-Loudness Level Contours". ISO 226, May, 1987.
- [JIN98] Just Intonation Network. <http://www.dnai.com/~jinetwk/>
- [Kashima85] Kashima, Mont-Reynaud. (1985). "The Bounded-Q Approach to Time-Varying Spectral Analysis". Stanford University, Dep. of Music Tech. Report STAN-M-23, 1985.
- [Kashino93] Kashino, Tanaka. (1993). "A sound source separation system with the ability of automatic tone modeling". Proceedings of the International Computer Music Conference, 1993.
- [Kashino95] Kashino, Nakadai, Kinoshita, Tanaka. (1995). "Application of Bayesian probability network to music scene analysis". Proceedings of the International Joint Conference on AI, CASA workshop, 1995.
- [Katayose89] Katayose, Inokuchi. (1989). "The Kansei music system". *Computer Music Journal*, 13(4), 1989.
- [Kay88] Kay. (1988). "Modern Spectral Estimation, Theory & Application". Prentice Hall.
- [Klassner95] Klassner, Lesser, Nawab. (1995). "The IPUS Blackboard Architecture as a Framework for Computational Auditory Scene Analysis". Proceedings of the Int. Joint Conference on AI, CASA workshop, 1995.
- [Koblitz87] Koblitz. (1987). "A Course in Number Theory and Cryptography". Springer, Berlin.
- [Kunieda96] Kunieda, Shimamura, Suzuki. (1996). "Robust method of measurement of fundamental frequency by ACLOS - autocorrelation of log spectrum". *IEEE Trans. on Acoustics, Speech and Signal Processing*, 1996.
- [Kuosmanen94] Kuosmanen. (1994). "Statistical Analysis and Optimization of Stack Filters". Tech.D. thesis., Acta Polytechnica Scandinavia, Electrical Engineering Series.
- [Ladefoged89] Ladefoged. (1989). "A note on 'Information conveyed by vowels'". *J. Acoust. Soc. Am.*, 1989.
- [Lahat87] Lahat, Niederjohn, Krubsack. (1983). "Spectral autocorrelation method for measurement of the fundamental frequency of noise-corrupted speech". *IEEE Trans. on Acoustics, Speech and Signal Processing*, No. 6, June 1987.
- [Laroche93] Laroche. (1993). "The use of the matrix pencil method for the spectrum analysis of musical signals". *J. Acoust. Soc. Am.*, Oct. 1993.
- [Lesser95] Lesser, Nawab, Klassner. (1995). "IPUS: An Architecture for the Integrated Processing and Understanding of Signals", *AI Journal* 77(1), 1995.
- [Maher89] Maher. (1989). "An Approach for the Separation of Voices in Composite Musical Signals". Ph.D. thesis, University of Illinois, Urbana-Champaign.
- [Maher90] Maher. (1990). "Evaluation of a Method for Separating Digitized Duet Signals". *J. Audio Eng. Soc.*, Vol. 38, No.12, 1990 December, p. 956.
- [Martin96a] Martin. (1996). "A Blackboard System for Automatic Transcription of Simple Polyphonic Music". MIT Media Laboratory Perceptual Computing Section Technical Report No. 399.
- [Martin96b] Martin. (1996). "Automatic Transcription of Simple Polyphonic Music: Robust Front End Processing". MIT Media Laboratory Perceptual Computing Section Technical Report No. 385.
- [McAdams89] McAdams. (1989). "Segregation of concurrent sounds. I: Effects of frequency modulation coherence". *J. Acoust. Soc. Am.*, Dec. 1989.
- [McAulay86] McAulay, Quatieri. (1986). "Speech analysis/synthesis based on a sinusoidal representation". *IEEE Trans. on Acoustics, Speech and Signal Processing*, 34(4), pp. 744-754, 1986.
- [Meillier91] Meillier, Chaigne. (1991). "AR modeling of musical transients". *IEEE Trans. on Acoustics, Speech and Signal Processing*, 1991, pp. 3649-3652
- [Mellinger91] Mellinger. (1991). "Event Formation and Separation in Musical Sounds". Ph.D. thesis, Stanford University. Dep. of Music Report STAN-M-77. 1991.
- [Miller75] Miller. (1975). "Pitch detection by data reduction". *IEEE Trans. on Acoustics, Speech and Signal Processing*, Feb. 1975, pp. 72-79.
- [Mont-Reynaud85a] Mont-Reynaud, Goldstein. (1985). "On Finding Rhythmic Patterns in Musical Lines". Proceedings of the International Computer Music Conference, 1985.
- [Mont-Reynaud85b] Mont-Reynaud. (1985). "Problem-Solving Strategies in a Music Transcription System". CCRMA, Dep. of Music, Stanford University.
- [Moore95] Moore (ed.). (1995). "Hearing. Handbook of Perception and Cognition (2nd edition)". Academic

- Press Inc.
- [Moorer75a] Moorer. (1975). "On the Transcription of Musical Sound by Computer". *Computer Music Journal*, Nov. 1977.
- [Moorer75b] Moorer. (1975). "On the Segmentation and Analysis of Continuous Musical Sound by Digital Computer". Ph.D. thesis. Dep. of Music, Stanford University.
- [Niihara86] Niihara, Inokuchi. (1986). "Transcription of Sung Song". Proceedings of the International Conference of Acoustics, Speech and Signal Processing, 1986, pp. 1277-1280.
- [Noll64] Noll. (1964). "Cepstrum pitch detection". *J. Acoust. Soc. Am.*, Feb. 1967.
- [Nunn94] Nunn. (1994). "Source separation and transcription of polyphonic music". <http://capella.dur.ac.uk/doug/icnmr.html>.
- [Opcode96] Opcode Studio Vision Pro. <http://www.dg.co.il/Partners/opcode.html>.
- [Ortmann26] Ortmann. (1926). "On the melodic relativity of tones". *Psychological Monographs*, 35, No. 162, 1926.
- [Pao89] Pao. (1989). "Adaptive Pattern Recognition and Neural Networks". Addison-Wesley Publishing Co.
- [Pawera81] Pawera. (1981). "Microphones: technique & technology". ARSIS Baedeker & Lang Verlags GmbH.
- [Piszczalski77] Piszczalski, Galler. (1977). "Automatic Music Transcription". *Computer Music Journal*, 1(4), 1977.
- [Qian97] Qian, Ding. (1997). "A phase interpolation algorithm for sinusoidal model based music synthesis". Proceedings of the International Conference on Acoustics, Speech and Signal Processing, 1997, pp. 451-454.
- [Rabiner76] Rabiner, Cheng, Rosenberg, McGonegal. (1976). "A Comparative Performance Study of Several Pitch Detection Algorithms". *IEEE Trans. on Acoustics, Speech and Signal Processing*, No. 5, Oct. 1976.
- [Rosenthal92] Rosenthal. (1992). "Machine rhythm: computer emulation of human rhythm perception". Ph.D. thesis, MIT.
- [Scheirer93] Scheirer. (1993). "Extracting expressive performance information from recorded music". MSc. thesis, MIT.
- [Scheirer95] Scheirer. (1995). "Using musical knowledge to extract expressive performance information from audio recordings". Machine Listening Group, MIT Media Laboratory.
- [Scheirer96a]. Scheirer. (1996). "Bregman's chimerae: music perception as auditory scene analysis". Machine Listening Group, MIT Media Laboratory
- [Scheirer96b]. Scheirer. (1996). "Tempo and Beat Analysis of Acoustic Musical Signals". Machine Listening Group, MIT Media Laboratory, 1996.
- [Schloss85] Schloss. (1985). "On the Automatic Transcription of Percussive Music - From Acoustic Signal to High-Level Analysis". Ph.D. thesis, Stanford University. Report STAN-M-27, 1985.
- [Serra89] Serra. (1989). "A system for sound analysis/transformation/synthesis based on a deterministic plus stochastic decomposition". Ph.D. thesis, Stanford University.
- [Serra97] Serra. (1997). "Musical Sound Modeling With Sinusoids Plus Noise". Roads, Pope, Poli (eds.). "Musical Signal Processing". Swets & Zeitlinger Publishers.
- [Slaney93] Slaney. (1993). "An efficient implementation of the Patterson-Holdsworth auditory filter bank". Apple Computer Technical Report 35, 1993.
- [Slaney95] Slaney. (1995). "A critique of pure audition". Joint International Conference on AI, CASA workshop, Aug. 1995.
- [Smith87] Smith, Serra "PARSHL: an analysis/synthesis program for non-harmonic sounds based on a sinusoidal representation". International Computer Music Conference, 1987.
- [Stautner82] Stautner. (1982). "The auditory transform". MSc. thesis, MIT.
- [Taylor93] Taylor, Greenhough. (1993). "An object oriented ARTMAP system for classifying pitch". International Computer Music Conference, 1993, pp. 244-247.
- [Thomassen82] Thomassen. (1982). "Melodic Accent: Experiments and a Tentative Model". *J. Acoust. Soc. Am.*, 71(6), 1982.
- [Todd92] Todd, McAulay. (1992). "The Auditory Primal Sketch: a Multiscale Model of Rhythmic Grouping". *Journal of New Music Research*, 23, pp. 25-70, 1992.
- [Warren70] Warren. (1970). "Perceptual Restoration of Obliterated Sounds". *Science* 167, 1970.