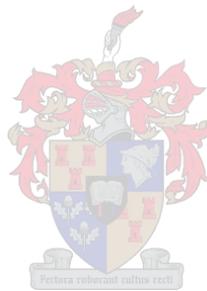


Automatic Music Transcription

An Exploratory Study

PETER E. MATTHAEI



*Thesis presented in partial fulfilment of the requirements for the degree
Master of Science in Electronic Engineering
at the University of Stellenbosch*

SUPERVISOR: Prof J.G. Lourens
CO-SUPERVISOR: Mr T. Herbst

April 2004

Declaration

I, the undersigned, hereby declare that the work contained in this thesis is my own original work, except where stated otherwise.

SIGNATURE

DATE

Abstract

In a pioneering project for the University of Stellenbosch, and indeed South Africa, an automatic music transcription system was designed to explore the underlying theory, concepts and problematics of polyphonic music transcription.

Automatic music transcription involves knowledge from the fields of acoustics, music theory, digital signal processing and information theory. The key concepts from these contributing fields as they relate to transcription systems are described in overview. A transcription system is then developed which includes components for FFT-based multi-pitch estimation, basic post-processing, estimation of the degree of polyphony, key determination, note duration quantisation and score output. The operation of the system is explained and tested at the hand of a synthetic polyphonic signal.

The system produced usable transcriptions of real monophonic input signals to scores with standard notational symbols. The success of the system (as are the successes of all published polyphonic transcription systems) was limited for real polyphonic music signals. Nonetheless, the initial results are encouraging and indicate that the current implementation can serve as a platform for a more sophisticated and accurate system.

Opsomming

In 'n baanbrekersprojek vir die Universiteit van Stellenbosch (en die breër Suid-Afrika) is 'n outomatiese musiek transkripsie stelsel ontwerp om die onderliggende teorie, konsepte en problematiek van polifoniese musiek transkripsie te ondersoek.

Outomatiese musiek transkripsie kombineer kennis uit die navorsingsvelde van akoestiek, musiekteorie, syferseinverwerking en informasieteorie. Die sluitelkonsepte van elkeen van hierdie velde word kortliks weergegee soos dit van toepassing is op transkripsie stelsels. 'n Transkripsie stelsel met modules vir FFT-gebaseerde afskatting van polifoniese toonhoogtes, basiese naverwerking, afskatting van die graad van polifonie, bepaling van die sleutel, nootlengte kwantisering en bladmusiek notasie word aansluitend ontwikkel. Die werkswyse van die stelsel word aan hand van 'n sintetiese polifoniese sein verduidelik en getoets.

Die stelsel lewer bruikbare transkripsies van enkelstemmige intreseine na bladmusiek met standaard musieksimbole. Die sukses van die stelsel is beperk vir polifoniese musiek, soos ook die algemene geval is vir ander gepubliseerde meerstemmige transkripsie stelsels. Tog is die aanvanklike resultate belowend, met aanduidings dat die huidige implementering kan dien as 'n beginpunt vir die ontwikkeling van 'n meer gesofistikeerde en akkurate stelsel.

Acknowledgements

I would like to thank the following:

- *Prof Johan Lourens* and *Mr Theo Herbst* for their academic guidance and moral support of this and other related music technology endeavours over the past two years.
- *Prof Johan du Preez* and *Mr Ludwig Schwardt* for ideas and programming advice.
- *Christoff Fourie*, *Theuns Louw* and *Emile de Roubaix* who performed the monophonic music samples which were used to test the system.
- *Wietsche Calitz* for his contributions to the original monophonic pitch tracking algorithm which was used as the basis for the multi-pitch estimator, as well as his knowledge of and enthusiasm for music technology.
- *The Sam Cohen Scholarship Trust* for financial support of my studies over the past six years.
- *The Harry Crossley Scholarship Fund* for financial support of my studies over the past two years.
- *All composers and musicians throughout the ages* without whom this fascinating field would not have existed.

And finally,

- *My long-suffering loved ones* for their love, patience and support in all things personal and academic.

Contents

1	Introduction	1
1.1	Motivation	1
1.2	Literature Synopsis	3
1.3	Objectives	3
1.4	Contributions	4
1.5	Overview of work	5
2	Literature Review:	
	Existing Solutions	7
2.1	A Brief History of Transcription	7
2.2	Transcription system considerations	8
2.2.1	Levels of representation	8
2.2.2	Computational vs. psycho-physiological models	8
2.2.3	System designs	9
2.3	Pitch estimation	12
2.3.1	Correlation-based methods	12
2.3.2	FFT	13
2.3.3	Constant Q	13
2.3.4	Bounded Q	15
2.3.5	Other	16
2.4	Wavelets	16
2.5	Neural Networks	17
2.6	Rhythm estimation	18
2.7	Instrument Identification	19
2.8	Key signature identification	19
2.9	Bayesian networks	20
2.10	Comparison of Different Systems	21

3	Basic Acoustics & Theory of Music	23
3.1	An overview of human audition	23
3.1.1	The human ear	23
3.1.2	Auditory streaming	25
3.2	Basic musical acoustics	26
3.2.1	Pitch	27
3.2.2	Timbre	29
3.2.3	Loudness	30
3.2.4	Superposition	32
3.3	Basic music theory	34
3.3.1	Notation	34
3.3.2	Diatonic Scales	36
3.3.3	Temperament	38
3.3.4	Meter and Rhythm	40
3.3.5	Polyphony defined	41
3.3.6	Harmony	42
3.4	Problems to be solved for complete music transcription	43
4	System Design	45
4.1	System components	45
4.1.1	Overview	45
4.1.2	Preprocessor	46
4.1.3	Extracting physical information from the waveform	46
4.1.4	Integrating the musical information	47
4.2	Restrictions	48
5	Multi-Pitch Estimation	51
5.1	Background	51
5.1.1	Importance and limitations	51
5.1.2	Requirements	51
5.1.3	The Basic Pitch Determination Problem	52
5.1.4	Mid-level representation	54
5.1.5	Synthetic signal	55
5.2	Pitch estimation algorithm	57

CONTENTS

iii

5.2.1	Algorithm outline	57
5.2.2	Determining spectral peaks	58
5.2.3	Determining pitch candidates	59
5.2.4	Candidate scoring	60
5.2.5	Candidate validation	60
5.2.6	Removing partials from the partial candidate list	61
5.3	Pitch tracking algorithm	62
5.3.1	Algorithm	63
5.3.2	Discussion	64
5.4	Limitations	65
6	Post-Processing	67
6.1	Introduction	67
6.2	Basic post-processing	67
6.2.1	Eliminating notes with low power	67
6.2.2	Detecting polyphony	68
6.3	Key Determination	70
6.3.1	Introduction	70
6.3.2	Algorithm	70
6.3.3	Complete algorithm	72
6.3.4	Discussion	73
6.4	Note duration quantisation	74
6.5	Overview	74
6.5.1	Algorithm	76
6.5.2	Example	76
6.5.3	Limitations	76
6.6	Output	77
6.7	Comments based on the synthetic signal	78
7	Implementation issues	80
7.1	Development of a music processing library	80
7.2	Speed	80
7.3	User-friendliness	81

8	Experimental Investigation	82
8.1	Introduction	82
8.2	Transcription of a monophonic recorder sample	83
8.2.1	Results	83
8.2.2	Discussion	84
8.3	Transcription of a monophonic violin sample	84
8.3.1	Results	84
8.3.2	Discussion	85
8.4	Transcription of a polyphonic organ sample	86
8.4.1	Results	86
8.4.2	Discussion	86
8.5	Transcription of a polyphonic piano sample	86
8.5.1	Results	86
8.5.2	Discussion	87
8.6	Observations	87
9	Conclusions and Recommendations	89
9.1	The story thus far	89
9.2	The road ahead	91
A	Pitch and Notation	99
A.1	Pitch conversions	99
A.2	Notation	99
B	LULU Smoothing & Peak Detection Algorithm	103
B.1	Introduction	103
B.2	Algorithm	104
B.3	Results	105
C	Frequency Sharpening With The Phase Vocoder	106
C.1	Introduction	106
C.2	Algorithm	106
C.3	Results	107
D	Lloyd-Max Quantisation	109
D.1	Introduction	109
D.2	Algorithm	109

CONTENTS

v

E The MIDI File Format

111

F The MusiX_TE_X Format

113

G Experimental Results

116

List of Figures

2.1	Levels of representation of a signal	8
2.2	Overview of selected transcription systems	10
2.3	Correlation-based pitch estimation methods	12
2.4	Time-frequency resolutions for various signal transforms	14
2.5	Triad classification network architecture	18
2.6	Overview of Klapuri’s meter estimation method	18
2.7	A possible taxonomy of orchestral instrument sounds	20
3.1	The human ear	24
3.2	Schematic view of the human hearing mechanism	24
3.3	Auditory Streams: Tracers and Watchers	26
3.4	Minimum durations for pitch sensation	28
3.5	Piano spectra for various notes	30
3.6	Superposition of two tones	33
3.7	Piano keyboard with notes, pitch and staff notation	34
3.8	Circle of fifths for key signatures	36
3.9	Frequency ratios in the diatonic major scale	38
3.10	Frequency ratios in the diatonic minor scale	38
4.1	Breakdown of the AMADEUS transcription system	50
5.1	Harmonic structure of synthetic “instrument” sound	56
5.2	Spectrogram of synthetic polyphonic sample	57
5.3	Peak picking on the spectrum of a synthetic polyphonic sound	59
5.4	Raw pitch points of synthetic polyphonic sample	62
5.5	Unprocessed pitch tracks of synthetic polyphonic sample	64
5.6	Singing sample with deep vibrato	65

6.1	Models of various scale types	71
6.2	Visualisation of key signature detection for the synthetic signal	74
6.3	Piano roll excerpt for the synthetic polyphonic sound	79
7.1	Hierarchy of the symbolic music data in the system	80
A.1	Keyboard with notation, pitch and sounding ranges	100
B.1	Example of a <i>LULU</i> -filtered spectrum	105
C.1	Instantaneous frequency of a sinusoid	107
C.2	Pitch track of sung scale when sinusoidal frequencies are sharpened	108
D.1	Partitioning of the real line into cells	109
F.1	MusiX _{TeX} Reference Chart	115
G.1	Steps during the transcription of a recorder sample	117
G.2	Piano roll plot of results for the recorder sample	118
G.3	Steps during the transcription of a violin sample	119
G.4	Piano roll plot of results for the violin sample	120
G.5	Steps during the transcription of a polyphonic organ sample	121
G.6	Piano roll plot of results for the organ sample	122
G.7	Steps during the transcription of a polyphonic piano sample	123
G.8	Piano roll plot of results for the piano sample	124

List of Tables

2.1	Published transcription systems	21
3.1	Pleasant Note Intervals	37
3.2	Construction of the C Major Scale	37
3.3	Equal tempered scales	39
3.4	The relationship between the harmonic series and pitch	40
3.5	Meter types	41
8.1	Accuracy of the recorder transcription	83
8.2	Accuracy of the violin transcription	85
A.1	Notes of scales based on C	101
A.2	Duration symbols	101
A.3	Typical time signatures	102

Nomenclature

Acronyms

ACF	Autocorrelation Function
AR	Auto-Regressive
ARMA	Auto-Regressive Moving-Average
DFT	Discrete Fourier Transform
FFT	Fast Fourier Transform
IDFT	Inverse Discrete Fourier Transform
IFFT	Inverse Fast Fourier Transform
MSE	Mean Squared Error
MUSIC	MUltiple SIgnal Classification
SACF	Summary Autocorrelation Function
STFT	Short-Time Fourier Transform

Symbols

θ	Phase angle [rad]
μ	Mean
\mathcal{A}	Set of harmonic amplitudes of a note
A	Amplitude
a_p	Model coefficient (pole p)
B	Tempo [bpm]
b_q	Model coefficient (zero q)
C	Pitch class
c	Pitch [cents]
e	Error signal
\mathcal{F}	Set of significant frequencies in a signal frame
f_0	Fundamental frequency, pitch [Hz]
f_h	Frequency of h -th harmonic [Hz]

NOMENCLATURE

f_{A4}	Standard pitch reference (constant) [Hz]
f_M	Frequency of note M [Hz]
f_{pitch}	Apparent pitch [Hz]
H	Total number of harmonics of a note
h	Number of a harmonic partial
I	Instrument (symbolic)
\mathcal{K}	Normalised harmonic structure of a sound
K_h	Normalised amplitude of h -th harmonic
L_{note}	Length of a note [number of analysis frames]
L_p	Sound pressure level [dB]
L_S	Perceived loudness of complex tone [sones]
l	Physical length [m]
M	Number of a semitone ($C_4 = 60$)
N_{FFT}	FFT length [samples]
N_P	Model order (poles)
N_Q	Model order (zeroes)
N_{win}	Window length [samples]
n	Sample number in a time series
$\mathcal{P}_{tot,cand}$	Total power in a candidates spectrum
$\mathcal{P}(f)$	Signal power at frequency f
$P(\bullet)$	Probability
p	Pressure [Pa]
p_0	Pressure at threshold of hearing ($p_0 = 20 \mu\text{Pa}$)
Q	Quality factor
\mathcal{R}	Set of note duration ratios
R_T	Ratio between note and beat duration
r	Radius [m]
S	Combination tone waveform
t	Time [s]
T	Tension
T_{beat}	Beat duration [s]
T_{note}	Note duration [s]
$U(C)$	Combination model used for finding key signatures
$V(C)$	Number of occurrences of C in data
$W(C)$	Result of circular convolution
x	Time signal
Y	Young's modulus

Chapter 1

Introduction

Like everything else in nature, music is a becoming, and it becomes its full self when its sounds and laws are used by intelligent man for the production of harmony, and so made the vehicle of emotion and thought.

Theodore Mungers

1.1 Motivation

Music is one of humanity's great achievements. It expresses our deepest desires, longings and sorrows, captures monumental events from our pasts, constructs elaborate fantasy worlds or portrays our weakest moments, using sound as a brush, pitch as paint, harmony to blend these together and rhythm to give it shape.

The advent of the information age has brought sophisticated tools with which music can be analysed and musical styles compared. However, automatic music transcription remains the proverbial "holy grail" of computer music analysis. Numerous factors contribute to this. It is a very broad field which involves and combines aspects of computer engineering and digital signal processing, music theory, physics and acoustics, and psychoacoustics and auditory perception, to name but a few disciplines that contribute crucial knowledge towards a solution. Automatic music transcription is a wonderful combination of the worlds of man and machine, the arts and the sciences.

Automatic music transcription is also one of the few fields in digital signal processing which is still wide-open. Few researchers have contributed to the field (compared to image and speech processing), and a comprehensive general solution to the problem, though seemingly ever lurking just across the horizon, has yet to be discovered.

This thesis is an attempt at creating a rudimentary yet functional automatic transcription system that can be used as the basis for further development. It also serves as a platform for stimulating interest in music technology, a fledgling field at our university.

The applications of music transcription are legion:

- *Music analysis:* Given an automatic music transcription system with a high degree of accuracy, experiments can be conducted to compare the styles of different composers, historical eras or compositional forms in terms of harmony, harmonic progression, and melody and rhythm structure, or even to analyse the expressive performance characteristics of various artists. Automatic music transcription would also be especially useful in analysing unnotated music (be it the many different forms of non-Western music which have thus far only been superficially examined, or the vast corpus of modern popular music which has also for the greater part not been exhaustively analysed).
- *Computer-assisted music instruction:* Automatic music transcription systems form the backbone of sight-singing tutors and aural training software (see [41]).
- *Sound separation:* Automatic music transcription is closely related to the field of sound separation of polyphonic music signals, since most transcription techniques strive to extract pitch and partial information from musical signals. This information could be used either to reconstruct individual parts separately, or to suppress certain parts in a mixture. Perhaps somewhat further afield, yet nonetheless relying on many of the same principles as music transcription, is the use of psychoacoustic properties and music analysis tools for the reconstruction of damaged or imperfect audio recordings.
- *Score generation for unnotated music:* This is probably the most obvious application of automatic music transcription. Although many forms of music, like Western classical music, are readily available as sheet music, it would nonetheless be convenient to be able to produce electronically editable and distributable scores. As mentioned above, there are also many valuable forms of music which have not been committed to paper yet.
- *Melody databasing:* An application which is becoming increasingly popular today is the use of melody databases to electronically store and retrieve specific melodies at will. Quite often, the retrieval can be achieved by whistling or humming a tune which is transcribed and compared to the melodies in the database. Although commercial software is already available to achieve this (see [40]), improvements such as automatic melody identification in polyphonic music would ease the input of such tunes.

1.2 Literature Synopsis

Automatic music transcription can be defined as “the act of listening to a piece of music and of writing down music notation for the notes that make up the piece” [38] or, more basically, as a problem of “allocating harmonics to notes and notes to instruments” [60].

The problem of automatic monophonic music transcription has been well researched and solutions have been documented for more than 25 years. Automatic music transcription, however, has only been solved for a very specific and very narrow set of musical signals. Generally even the best transcription systems struggle with degrees of polyphony greater than four, and most transcription systems place severe limits on the specific mixtures of pitches or the specific mixtures of instruments that can be transcribed.

A number of techniques have been proposed to deal with various subproblems within the broad field of automatic transcription. Multi-pitch estimation is at the heart of automatic music transcription: finding the component pitches in a mixture of different notes sounded at the same time. Methods based on autocorrelation which model certain aspects of human audition seem to be the current trend in multi-pitch estimation. Many classic music transcription systems made use of sinusoid tracks, which can be most conveniently analysed with frequency domain methods such as the Discrete Fourier Transform (DFT) and Constant Q and Bounded Q analysis. Other techniques which have been explored by various researchers include wavelet analysis and neural networks.

A problem that is generally solved in parallel with multi-pitch estimation is rhythm analysis which strives to detect time periodicities in the music. A number of techniques have been reported in literature which address this problem fairly successfully, either by investigating the intervals between the onsets of note pairs, or (perhaps with more accuracy) by detecting periodicities in the time envelope of musical waveforms.

Other subproblems that have been addressed in literature include the identification of instruments and the construction of instrument models to aid in multi-pitch estimation, key signature identification, and the use of higher level musicological models to enhance the accuracy of various steps in the music transcription process. Chapter 2 discusses all of these issues in greater depth.

1.3 Objectives

The main objective of the present thesis is to implement a functional automatic music transcription system which takes acoustic signals (in *wav* files) as input and produces a transcription using an appropriate symbolic representation. The system should have at least the following properties:

- As an *automatic* system, it should require minimal human intervention.
- It should be able to transcribe music accurately to the extent that an audio reconstruction of the transcription output should be recognisable as a representation of the original acoustic signal.
- The system should be able to detect the key of the input signal accurately.
- For monophonic input signals, the symbolic output should be in the form of a MIDI file as well as in MusiXTEXscore representation.
- For polyphonic input signals, the symbolic output should be in the form of a MIDI file.

Secondary objectives of this study include the following:

- As this is one of the first large-scale music technology projects undertaken by our faculty, the basic concepts underlying human audition, auditory scene analysis, musical acoustics and music theory have to be explored.
- The present thesis should be able to serve as a platform for further research into the field of computer music analysis. To this end, a library of usable functions has to be developed which can be used in future music processing software.

It can hardly be stressed enough that this thesis is mostly exploratory in nature, being the first automatic music transcription project of this scale undertaken on the African continent. Existing published solutions are generally the fruit of many years of dedicated research. It would be unrealistic to expect to turn a tone-deaf computer into a Mozart in the span of mere months. Thus the objective was *not* to aim for a phenomenal and unprecedented transcription success rate, but rather to design and test a simple system that serves to highlight certain issues of automatic transcription. It is also noteworthy that this thesis attempted to design components at virtually every processing level, and thus provides an overview of the field in (virtually) its entirety. In that sense, this thesis is a *journey* through the rough seas of automatic music transcription, and not a *destination* in itself. The resulting system should similarly be seen as the embryonic genesis of an on-going research project, as opposed to providing closure on the issue.

1.4 Contributions

Following contributions were made to the automatic transcription field at large:

- A heuristics-based frequency-domain multi-pitch estimator was developed, which includes a powerful non-linear spectral smoothing algorithm for detecting spectral peaks.

- A number of useful algorithms were developed which can be implemented in other transcription systems to enhance their accuracy. These include algorithms for the elimination of soft notes, key identification, estimation of the degree of polyphony and note duration quantization.
- The transcription system produces MIDI and written score output for monophonic input signals that are sufficiently accurate so that they can be read and edited by musicians.
- Although the success of the system with polyphonic input signals is moderate due to the lack of incorporation of higher-level musicological information, the output is nonetheless acoustically similar to the input. This suggests that the approach followed in designing the system has merit. This is further underlined by the success of the system when applied to synthetic polyphonic signals.

1.5 Overview of work

Firstly the successes, limitations and architectures of existing transcription systems, as well as various techniques that have been applied to subproblems, will be described in the literature review in Chapter 2. The fundamental concepts of acoustics and theory of music are then examined in Chapter 3 to determine the basic building blocks of music that need to be dealt with by a transcription system, as well as to determine possible solution strategies.

After the literature overview, the architecture of the system is discussed in Chapter 4 in terms of the design philosophy and the system components. Chapter 5 will build up the theory for a frequency-domain based multi-pitch estimator and pitch tracker. The following chapter then explores the further processing of the data from raw pitch tracks to symbolic output in MIDI format (for monophonic and polyphonic music signals) and MusiX_{TEX} score output (for monophonic music signals). Algorithms are proposed to eliminate spurious notes, detect the key signature and scale type (major and minor), estimate the degree of polyphony and quantise the note durations to appropriate musical symbols. All the algorithms are explained and tested with a synthetic music signal.

Chapter 8 documents signals at various stages of the transcription process, from raw waveforms to piano rolls (and score output for monophonic music). Four signals are tested and the system performance discussed. The samples are, in increasing order of complexity:

- a slow monophonic recorder sample,
- a fast monophonic violin sample,

- a polyphonic organ sample, and
- a polyphonic piano sample.

In the concluding Chapter 9, the system performance is summarised and recommendations are made for further research. These recommendations include the implementation and use of

- music knowledge sources at various levels of processing to enhance the accuracy of the system,
- top-down information flow so that data extracted from higher levels can be used to increase the processing accuracy at lower levels,
- a robust rhythm and meter analysis component to allow for time signature identification, measure segmentation, and alignment of processing frames with metric events, and
- musical instrument models to increase the accuracy of multi-pitch estimation.

Chapter 2

Literature Review: Existing Solutions

2.1 A Brief History of Transcription

Gerhard provides a good and gentle introduction to computer music analysis, with particular emphasis on automatic music transcription [20]. Klapuri, Martin, Scheirer and Walmley also provide information surrounding the development of the field [28, 38, 55, 60].

One of the most comprehensive solutions to monophonic music transcription has been given by Martin Piszczalski, with later modifications by Galler [20], where the mid-level representation of the audio signal was the FFT. Today there exists a plethora of systems that can perform monophonic music transcription, to varying degrees of success.

James Moorer provided the first documented polyphonic music transcription system, for two voices. The restrictions were very severe, such that the instrument sounds had to be piecewise constant (no vibrato, glissando, etc.), and most significantly, that the pitch and harmonics were not allowed to overlap [20, 28]. Moorer's system was later slightly improved upon by Maher, with the restriction that instruments had to occupy mutually exclusive pitch ranges [28, 38].

Hawley in 1993 reported a system that was capable of transcribing piano music quite accurately, based on differential spectra obtained with the FFT [38].

The current champions on the transcription scene are systems developed independently at Tokyo University and MIT. Both systems are reported to transcribe 3 and 4 voice polyphonies quite reliably [27, 28, 38].

However, in spite of some improvements in the reliability of transcription systems, partly due to more sophisticated architectures and the implementation of (still limited) top-down processing, the generality of current transcription systems is still severely restricted in terms of range, instrumentation and polyphony.

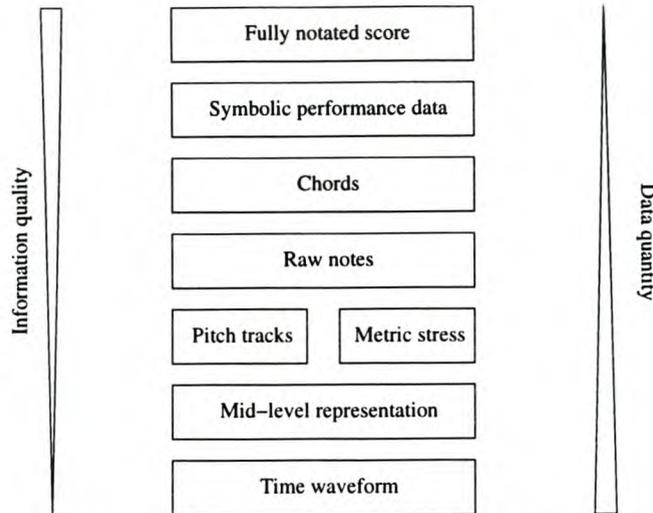


Figure 2.1: *Levels of representation of a signal*

2.2 Transcription system considerations

2.2.1 Levels of representation

The levels of representation of a music signal are shown in Figure 2.1. The first (bottom) level of representation is the sampled waveform. Although some processing techniques (like pitch detection by counting zero crossings) make use of this first level of representation, the signal is generally transformed to a different level of representation to allow more convenient processing of certain features of the signal [28]. Mid-level representation is generally either time-domain based with the correlogram (which allows more directly for periodicity analysis) or frequency-domain based with the spectrogram (which allows more directly for harmonicity analysis), although a technique like the wavelet transform combines features of both domains. The advantages and uses of different mid-level representations will be discussed in Section 2.3. The desired top-level representation of a successful transcription system would be a musical score in standard music notation. It is however fiendishly difficult to transform pitch tracks into a complete and accurate performance score, and thus many transcription systems opt for writing their output to MIDI files (the “symbolic performance data” level) instead [20].

2.2.2 Computational vs. psycho-physiological models

The human auditory perception system is an example of a working system capable (given sufficient talent, experience and training) of transcribing very complex musical pieces. Thus it can be argued that transcription systems should try to simulate the biological

and neuro-biological properties of hearing and perception. The study of human auditory perception is undeniably a vital contributing field to automatic music transcription. However, trying to comprehensively and consistently mimic the characteristics of the human ear to provide mid-level representations of musical signals seems senseless at best. Our understanding of human auditory perception is still very sketchy (hence the ongoing debate as to the validity and relationship of the *place* vs. *periodicity* theories of hearing¹) [20]. Thus techniques like the cochleagram, which tries to mimic the “filter bank” of the cochlea and is thus valuable and interesting from a perception modelling perspective, are quite often clumsy and cumbersome from a digital signal processing perspective when applied to pitch detection.

Another very important consideration in this regard is the fact that

[...] music often tries to fool the auditory system into hearing fictional streams.

[...] In order to get sounds to blend, the music must defeat the scene-analysis processes that are trying to uncover the individual sources of sound. [3, p. 457]

This implies that a transcription system that tries to mimic human audition too closely will similarly be fooled into observing “fictional streams”. Successful transcription thus needs to incorporate some knowledge of the individual sources in order to “reverse-engineer” the sound mixture into the *physical* sources (as opposed to the *perceived* sources).

2.2.3 System designs

Different researchers have used slightly different approaches in designing and integrating the different components of their transcription systems. Figure 2.2 shows block diagrams of two of the most recent published transcription systems in (a) and (b), along with Martin’s more recent processing front-end in (c) and Klapuri’s system break-down in (d). All diagrams are based on figures by the original authors, as cited in the captions.

The OPTIMA (Organised Processing Toward Intelligent Music Scene Analysis) system of Kashino *et al.* depicted in Figure 2.2(a) is perhaps the most advanced system yet implemented and published, given its strong emphasis on knowledge integration and top-down processing. This greatly enhances the accuracy of transcription results, as reported by the authors [27].

Martin’s initial blackboard system (Figure 2.2(b)) is also fairly sophisticated in that it integrates numerous knowledge sources and implements processing rules to form hypotheses at each layer of the data hierarchy. Figure 2.2(c) shows Martin’s more recent

¹These concepts will be elaborated on in Section 3.1.

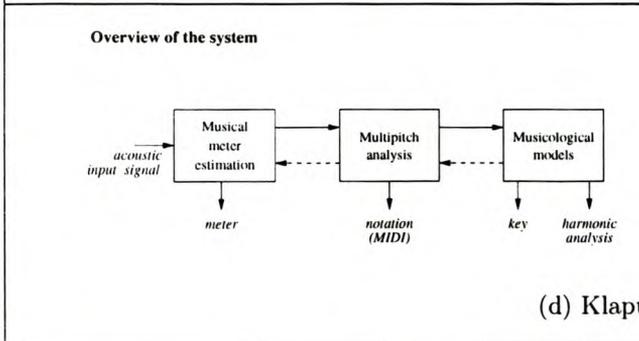
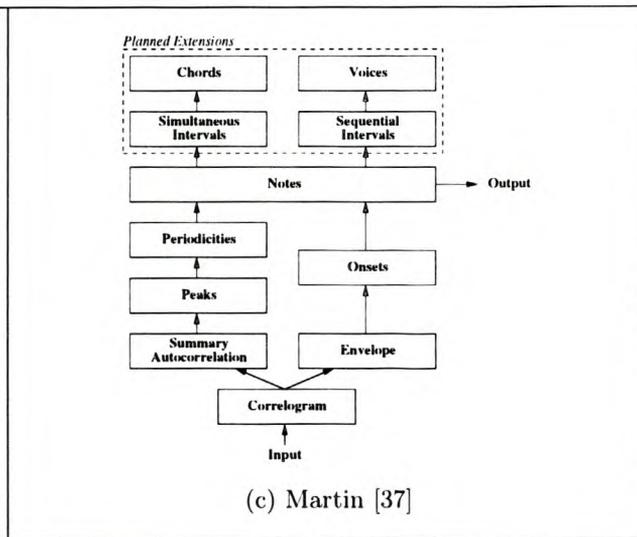
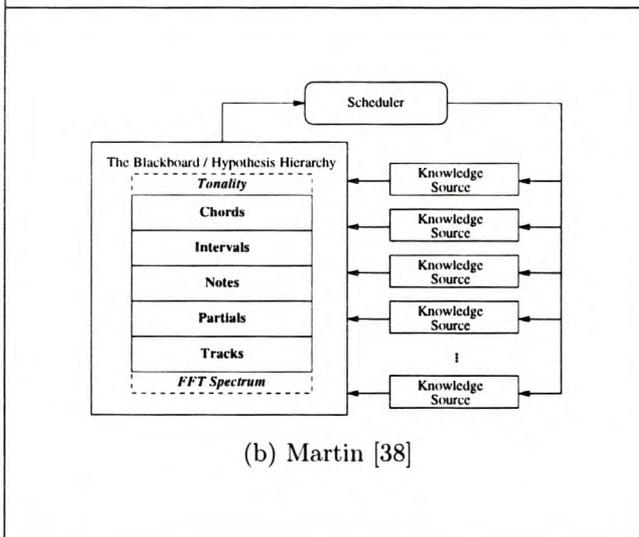
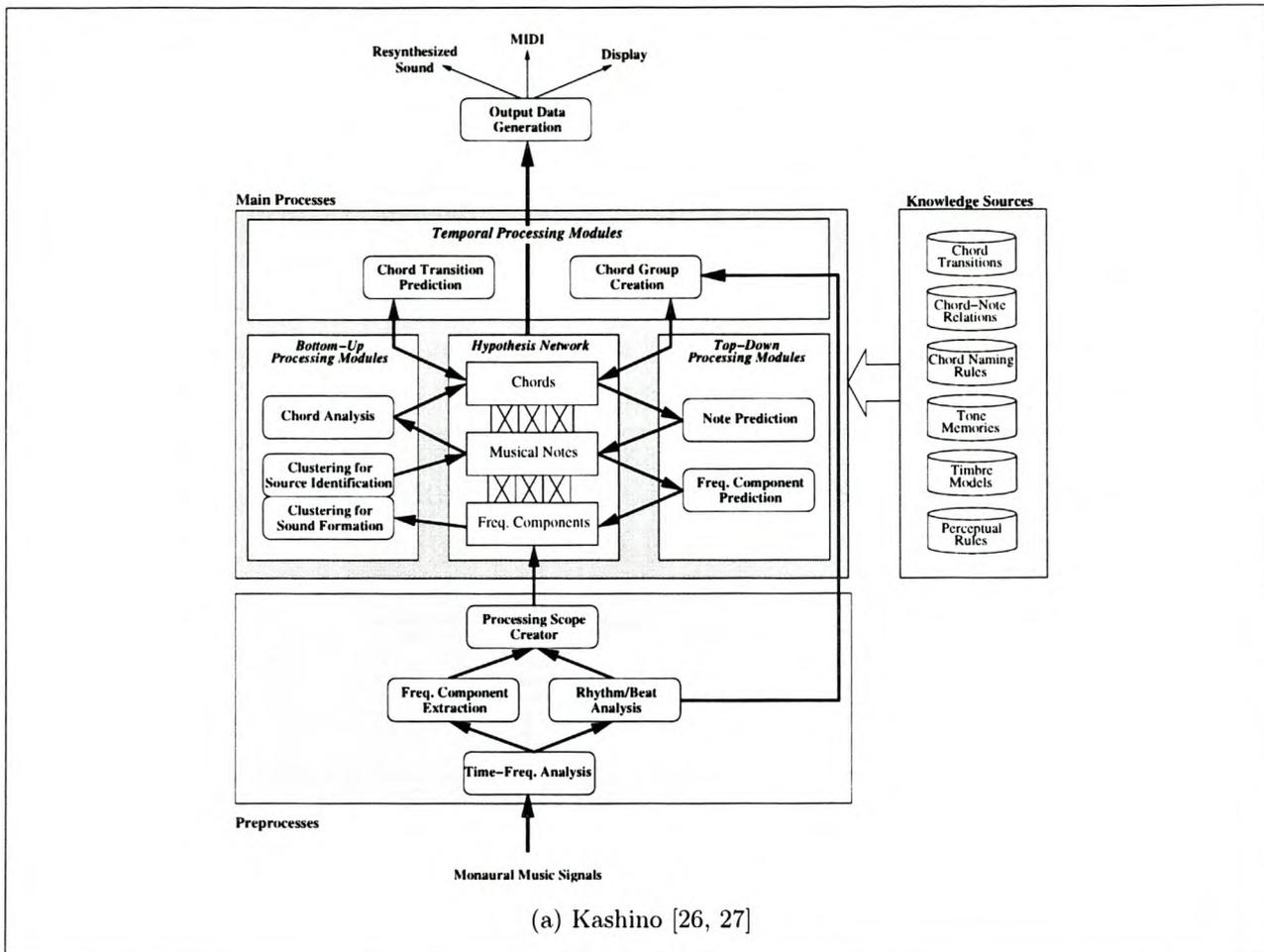


Figure 2.2: Overview of selected transcription systems

processing front-end for multipitch estimation which is correlogram-based, unlike his earlier FFT-based front-end.

The following are the most important conclusions that can be drawn from a study of various systems and the literature from which they are taken:

- *The necessity of a multi-pitch estimator and a rhythm/beat analyser:* Transcription systems require a component which can detect note onsets and a component which can estimate polyphonic pitch. The rhythm analysis can be used to align the multi-pitch estimator with note onset events.
- *The importance of using knowledge sources:* Without making use of any external knowledge about the mixture signal, the transcription problem remains an unsolvable single linear equation in multiple unknowns (see Equation 3.18). Martin lists three categories of knowledge sources: knowledge about *human auditory physiology*, knowledge about the *physics of sound production* and knowledge about the *rules and heuristics governing tonal music* [38]. Each of these knowledge sources provides a wealth of information that can be used by various steps in the transcription process.
- *The use of top-down processing:* The system diagrams of Kashino and Klapuri make reference to top-down processing in addition to bottom-up processing. In bottom-up processing, the flow of data is exclusively from the raw input signal towards the final output. In top-down processing, results from components higher up in the processing hierarchy are used as an additional input to lower levels to refine the results with expectations about the data. For example, if the tonality of a piece becomes known after some processing, the grouping of partials into notes can be assisted with rules which favour certain combinations of notes above others. The use of top-down processing is equivalent to the way in which humans tend to uncover more and more depth in a piece of music with each listen because familiarity with the music creates knowledge and expectations which help them to “look out” for events like chord resolutions or key changes. In effect, top-down processing introduces a dynamic knowledge source into the system which uses knowledge extracted from the signal itself.
- *The enormous scope of the problem:* OPTIMA is reported to consist of 60 000 lines of C-code (excluding the GUI) [27], and its solution to the transcription problem is still far from being generally applicable! Although improvements have since been made, Martin’s initial system was restricted to a range of B₃ to A₅ and required that no two notes ever be played simultaneously an octave apart [38]. The consensus amongst researchers seems to be that a complete solution is still a way off.

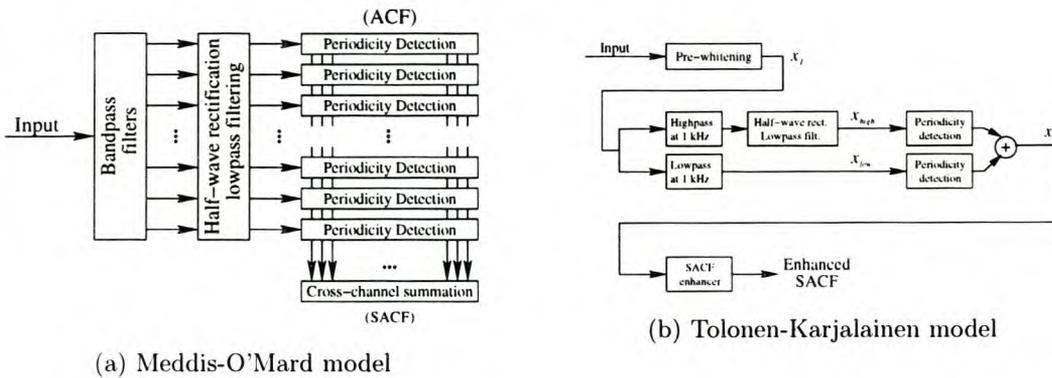


Figure 2.3: Correlation-based pitch estimation methods (based on [58])

2.3 Pitch estimation

2.3.1 Correlation-based methods

Correlation-based pitch estimation methods have found widespread use in monophonic pitch tracking [41, 42]. However, most early polyphonic pitch estimation methods were frequency-domain (DFT/Constant Q) based [20]. In fact, in his M.Sc. thesis, Klapuri noted that “utilisation 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” [28]. However, ever since Martin’s 1996 paper in which he proposed the use of log-lag correlograms for pitch estimation [37], a shift seems to have occurred towards autocorrelation-based methods. These models strive to emulate human auditory perception from a *periodicity theory* point-of-view and are less reliant on instrument models for octave detection and sound separation [37, 58].

In [25] and [58], two correlation-based methods are discussed: the multi-channel pitch analysis method of Meddis and O’Mard, and a two-channel method that has some similarities with the former. A block diagram of the Meddis-O’Mard model is given in Figure 2.3(a). The input is first divided into a number of bands corresponding roughly with the bandwidth channels of human audition. Each band is subsequently half-wave rectified and lowpass filtered to provide the signal envelope in each of the corresponding bands. Each band signal is then autocorrelated to detect periodicities, and all bands summed to produce the Summary Autocorrelation Function (SACF) which provides a measure of the overall periodicities in the signal.

The method developed by Karjalainen and Tolonen reduces the multi-channel approach to two channels. Their approach is visualised in Figure 2.3(b). The signal is

pre-whitened with a warped linear prediction filter to remove short time periodicities in the signal. The signal is then split into a high frequency (> 1 kHz) component and a low frequency component (< 1 kHz). Generalised autocorrelations (calculated from the IDFT of a magnitude compressed DFT of the signal) are calculated for the low frequency component and the *amplitude envelope* of the high frequency component respectively, and summed to provide the SACF. This division of the signal is chosen to emulate the neural firing in human audition (which exhibits direct time synchrony for low frequencies and envelope-based synchrony for higher frequencies).

The SACF will typically provide strong peaks for the common root of chords, and does not in itself provide clear peaks for the individual component notes. For this reason, the Enhanced SACF is calculated from the SACF. The SACF is clipped to positive values and then time-stretched by a factor 2. The stretched signal is then subtracted from the original clipped SACF to remove repeated peaks at double the time lag. The process is repeated by time-stretching by factors of 3, 4, 5, etc. For component note sounds with similar amplitudes, the component pitches can be determined from the remaining prominent peaks.

Tolonen and Karjalainen report promising results for this multi-pitch estimation method, provided that the component sounds have similar amplitudes and have fundamental frequencies less than the 1 kHz band cross-over frequency. They suggest that for mixtures of tones with dissimilar amplitudes, the tones with higher amplitudes should first be detected and filtered out (using, for example, a comb filter) so that the softer tones can be detected in the residual signal.

2.3.2 FFT

Of all frequency-based analysis methods, the Discrete Fourier Transform (DFT) is probably the most widely used in digital signal processing, and any DSP programming toolkit typically contains a number of computationally efficient implementations. The Short-Time Fourier Transform (STFT) is characterised by both constant time and constant frequency resolution, and assumes (as do most other analysis techniques) that the signal is stationary for the duration of an analysis frame. The FFT is discussed in more depth in Chapter 5 in the context of multi-pitch estimation.

2.3.3 Constant Q

In order to address the time/frequency resolution trade-off of the FFT, Brown and others have developed a spectral analysis method with a constant Q [4]. The quality factor Q is defined as the ratio of center frequency to frequency resolution for a specific spectral

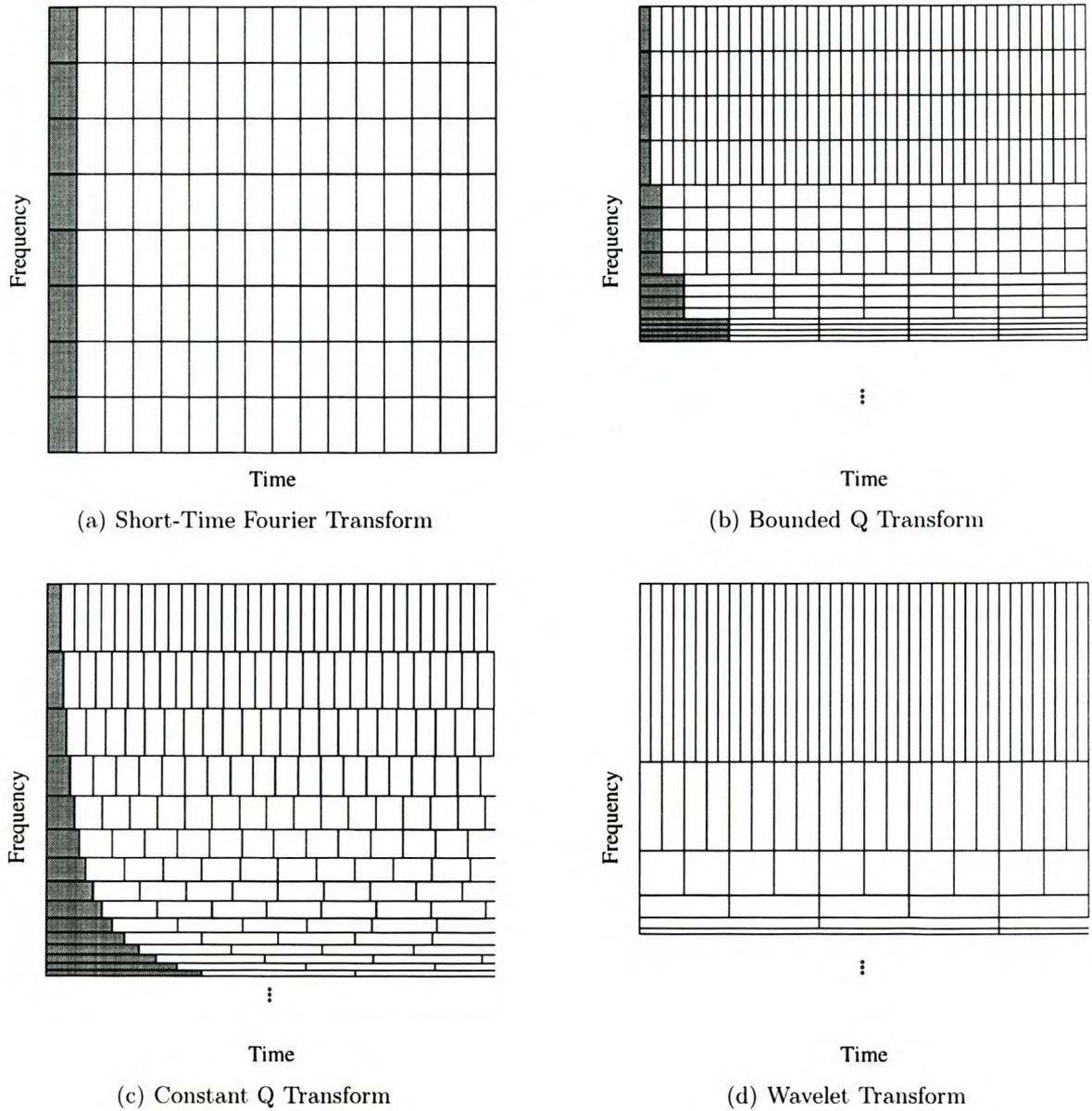


Figure 2.4: *Time-frequency resolutions for various signal transforms. The shaded tiles are those which constitute one analysis frame.*

component f_k :

$$Q = \frac{f_k}{\Delta f_k} \quad (2.1)$$

The Constant Q transform, visualised in Figure 2.4(c) with four bins per octave, is conceptually equivalent to an FFT with logarithmic frequency bins. Thus the Constant Q transform has the advantage that lower notes can be resolved equally well as higher notes, because the CQ bin centre frequencies can be chosen to coincide with the semitones (or quarter tones) of the equal-tempered² chromatic scale. Another advantage of the logarithmic frequency axis is that the harmonics h of all notes bear the same relationship to the fundamental, due to:

$$f_h = hf_0 \quad (2.2)$$

$$\Rightarrow \log f_h = \log hf_0 \quad (2.3)$$

$$= \log h + \log f_0 \quad (2.4)$$

Brown proposes a technique whereby the Constant Q transform can be calculated from the FFT by straightforward multiplication of the FFT with frequency domain kernels which represent the way each FFT bin contributes to a specific Constant Q bin [5].

Constant Q analysis has found widespread use in computer music analysis and automatic music transcription due to its convenient relationship to the properties of equal-tempered scales and instrumental harmonics.

2.3.4 Bounded Q

Instead of Constant Q analysis, researchers at Stanford have proposed a Bounded Q transform which is essentially an STFT analysis on an octave-by-octave basis [4, 6, 28]. The FFT is calculated for a signal frame. All frequency-domain values except for the highest desired octave are then discarded. The signal frame is subsequently filtered and downsampled by a factor of 2. Another FFT operation with the same number of points is performed on the downsampled signal frame. The new FFT thus has twice the resolution of the first one. This time only frequency-domain points in the second-highest octave are kept. The process is repeated until the lowest octave desired is reached. The Bounded Q Transform thus gives a constant number of frequency-domain samples (typically around 32) per octave. The time-frequency resolution of the Bounded Q Transform is visualised in Figure 2.4(b) for four bins per octave.

²Defined in Section 3.3.3.

2.3.5 Other

Walmsley [60] discusses a variety of other methods for spectrum estimation in the context of music analysis. These methods include filter-based AR and ARMA models (AutoRegressive/AutoRegressive Moving-Average methods) which analyse data by considering each sample as part of a time series and finding models and associated parameters to “predict” the next sample $x[n]$. The AR model is an all-pole model, corresponding to an IIR filter driven by white noise e . For a model of order N_p with coefficients a , $x[n]$ is calculated as:

$$x[n] = \sum_{p=1}^{N_p} a_p x[n-p] + e[n] \quad (2.5)$$

ARMA models have poles and zeros and, being more general, can thus model signals using fewer parameters. ARMA models are given by:

$$x[n] = \sum_{p=1}^{N_p} a_p x[n-p] + \sum_{q=1}^{N_q} b_q e[n-q] \quad (2.6)$$

where N_q gives the number of zeros and b are the corresponding zero coefficients. The poles and zeros of AR and ARMA models can be evaluated on the discrete unit circle to provide spectral estimates.

Another method which can be used for (pseudo-) spectrum estimation is MUSIC (Multiple Signal Classification), which is a signal subspace method. Subspace methods can resolve closely spaced frequencies due to their good frequency resolution. However, if the model order is chosen inappropriately, MUSIC can give spurious frequency components. Furthermore, MUSIC gives (theoretically) infinite peaks at sinusoids and can thus not be used for estimating the amplitudes of the harmonics.

2.4 Wavelets

A technique related to Constant Q analysis is the wavelet transform, a multi-resolution analysis technique which trades off time resolution for greater frequency resolution at low frequencies (see Figure 2.4(d)). Cemgil describes a wavelet theoretical approach to monophonic music transcription in his M.Sc. thesis [6]. A basis function that is matched to the properties of the signal is used: The wavelet basis function is a linear combination of complex exponentials to simulate the harmonic nature of musical sounds (hence the term “matched basis function”). The advantage of this approach is that if the analysis is done in the frequency band where the signal is expected, energy information from higher harmonics can be combined to the analysis band.

Another wavelet-based monophonic pitch tracking approach is documented in [17], which finds the locations of local maxima at various scales in a wavelet-transformed signal and calculates the pitch as the time distance between two consecutive maxima. This is the wavelet-domain equivalent of pitch determination by counting zero-crossing.

2.5 Neural Networks

A few researchers have investigated the use of neural networks in computer music analysis. Neural networks are typically higher level processing techniques which take as input some mid-level representation like Fourier-transformed or wavelet-transformed data.

Klingseisen and Plumbley [31] have applied neural nets to musical instrument separation making use of the multiple cause model first described by Saunders. Unlike most other neural nets which are “winner-takes-all” techniques which can account for only a single cause, multiple cause models try to take all underlying causes into account and are thus well suited for analysing mixtures and separating the causes. Good results were reported for simple synthetic mixtures of artificial spectra. Real instruments however have a number of features which complicate an analysis with neural networks: Different notes played on the same instrument have different spectra, and even the *same* note played at different intensities has different spectra. Training the network with all possible combinations of different notes, instruments and volumes may prove a major hurdle in the success of this technique.

Shuttleworth and Wilson [56] initially took a slightly different approach in their application of neural networks to music analysis: instead of trying to separate musical instruments, they tried to detect and classify musical triads. Figure 2.5 shows the network that they used. However, they reported an asymptotic successrate of only 53% in their experiments which is “almost certainly too low a rate to be of direct use in a transcription system” [56]. They then proceeded to investigate the use of neural networks in note recognition using the following linear model:

$$\mathbf{y} = H\mathbf{x} + \mathbf{n} \tag{2.7}$$

where \mathbf{x} contains impulses at the elements representing active pitch positions which are transformed into the “blurred” spectrum \mathbf{y} by the spectral “blurring” matrix (where each column represents the template spectra for each note), with an additive noise term \mathbf{n} for inharmonic and incidental noise. They proceed to construct a neural network to find \mathbf{x} (given \mathbf{y} and an approximate H) by minimising an error function. This approach contains an interesting perspective on the transcription problem, inspite of the relatively poor reported successful recognition rate (38%) of all three notes in tested musical triads.

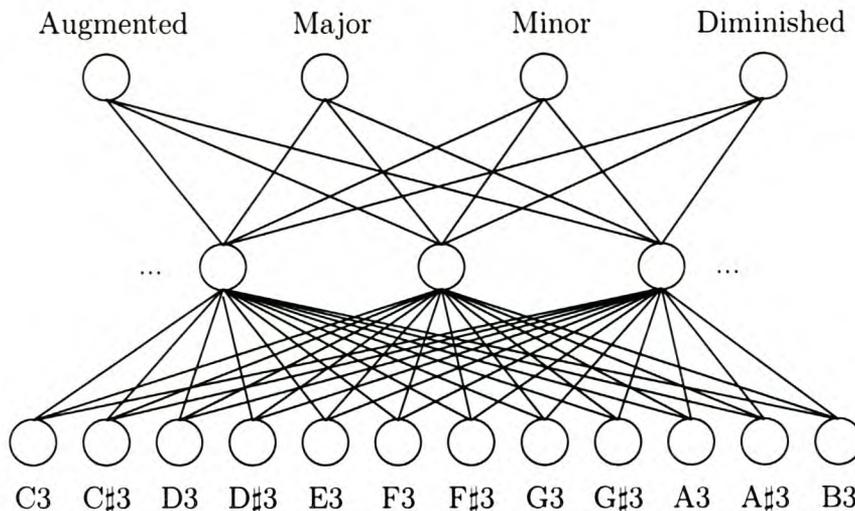


Figure 2.5: *Triad classification network architecture (based on [56])*

2.6 Rhythm estimation

Rhythm estimation is a parallel task to pitch tracking, and is often treated separately as a different specialised area. Rhythm is generally divided into three metrical levels, namely the *tatum* (the time quantum or smallest note duration of which all other durations are integer multiples), the *beat* (foot-tapping rate) and the *measure*. These three values are interrelated and can be used together as a feature vector [30].

For automatic music transcription purposes, it is desirable to uncover temporal periodicities directly from the acoustic input. Klapuri describes a generalised algorithm to find registral accent signals $v_c[n]$ (the degree of accentuation as a function of time) in various frequency bands [30]. These signals are calculated from the smoothed power envelopes and their derivatives in various frequency bands of the acoustic signal and thus express at each time instance the degree of loudness and change of loudness in the input signal.

Periodicities in the registral accent signals can then be estimated by a variety of methods, such as enhanced cross-correlation, phase-locking resonators and comb-filter

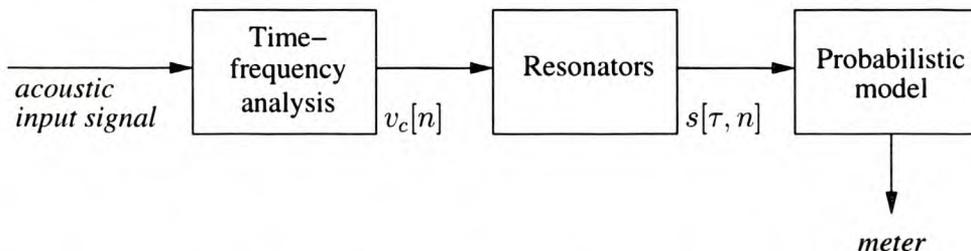


Figure 2.6: *Overview of Klapuri's meter estimation method (based on [30])*

resonators. A function $s[\tau, n]$ of the strength of different metrical pulses at each time instant is used as the input to a probabilistic meter estimation algorithm. In order to allow the meter to change over time, an HMM-type state machine is used to determine the most likely development of the meter over time.

Klapuri describes promising results with this method which was developed as a synthesis of techniques by a number of other researchers. An overview of this method is depicted in Figure 2.6.

A number of techniques for rhythm analysis use symbolic input in the form of MIDI files instead of acoustic waveforms from which they perform their periodicity analysis. Dixon describes a beat tracking technique based on the scoring of inter-note onset intervals $\Delta t = t_2 - t_1$ between pairs of notes which are not separated too far in time. The beat interval is estimated as the value which best accounts for the set of inter-onset intervals [15].

Cemgil describes a tempo tracking system which makes use of Kalman filtering. The tempo tracker is modelled as a stochastic dynamical system where the tempo is a hidden state variable which can be estimated with a Kalman filter [8].

2.7 Instrument Identification

Martin [39] describes a possible approach to instrument identification. For each instrument in a sample set, 31 features are extracted, including pitch variance, tremolo frequency and strength, average spectral centroid, odd/even harmonic ratio, vibrato amplitude and frequency, and onset duration. Instruments are then classified down an instrument taxonomy, as shown in Figure 2.7: first the instrument *family* is recognised using the feature vector, and then the *instrument* itself is recognised within that family. This is, according to Martin, based on common human experience: first we recognise a certain sound as being, say, produced by a bowed string instrument before we recognise it as either a violin or viola sound. He reports accuracies of upwards of 70% for classifying individual instruments in a set of 14 orchestral instruments.

2.8 Key signature identification

The importance of a key-finding algorithm is addressed by Krumhansl, as follows:

For automatic music analysis of tonal music, the key needs to be determined in order for the structural roles of melodic and harmonic events to be coded meaningfully. For example, in connection with harmonic analysis, Winograd (1968) noted the inherent ambiguity of chords and the necessity

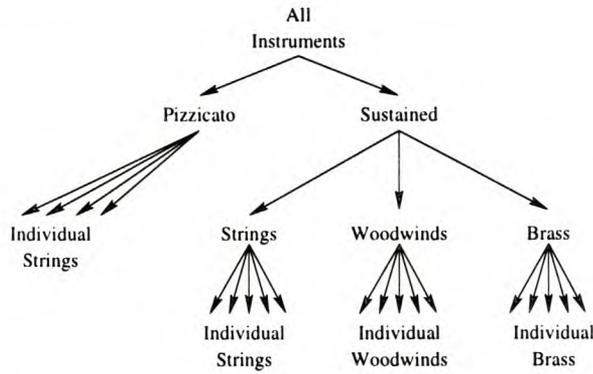


Figure 2.7: A possible taxonomy of orchestral instrument sounds (based on [39])

to ascribe meaning to them in terms of their functions within the system of interrelated tonalities. [33, p. 77]

Identifying the key signature of a performance is addressed briefly in [30]. The method described uses Bayes' formula

$$P(k|m) = \frac{P(m|k)P(k)}{P(m)} \quad (2.8)$$

to estimate the probability of the occurrence of a note's pitch class m in the key k . For a sequence of notes $\mathcal{M} = \{m_1, m_2, \dots, m_T\}$, the probability of key k is given by the product of the Bayesian probabilities for each note:

$$P(k|\mathcal{M}) = \prod_{t=1}^T P(k|m_t) \quad (2.9)$$

The probabilities $P(m|k)$ can be calculated from the table of tonal distributions in [33, p. 67].

2.9 Bayesian networks

A number of researchers have used Bayesian probability models for various purposes:

- Kashino *et al.* made extensive use of Bayesian networks for information integration in their system [26, 27]. Their network has three layers: a component level, a note level and a chord level. The component level is connected to the note level with a single link (a link which corresponds to one temporal processing step). The note level is connected to the chord level with a multiple link (a link which connects notes from possibly several temporal processing steps due to the fact that multiple notes along the time axis may form a single chord). The various chord notes are connected to each other with a temporal link which encodes chord progression. All the links use Bayesian probabilities to formulate and score various hypotheses.

Table 2.1: *Published transcription systems (based on [28])*

<i>Main Author</i>	<i>Institute</i>	<i>Polyphony</i>	<i>Range</i>	<i>Knowledge</i>
Moorer 1975	Stanford University	2	severe limitations on content <i>Sounds:</i> Violin, guitar	24 Heuristic approach
Chafe 1982	Stanford University	2	presented simulation results insufficient <i>Sounds:</i> Piano	19 Heuristic approach
Maher 1989	Illinois University	2	severe limitations, pitch ranges must not overlap <i>Sounds:</i> Clarinet, bassoon, trumpet, tuba, synthesised	Heuristic approach
Katayose 1989	Osaka University	5	several errors allowed <i>Sounds:</i> Piano, guitar, shamisen	32 Heuristic approach
Nunn 1994	Durham University	≤ 8	several errors allowed, perceptual similarity <i>Sounds:</i> organ	48 Perceptual rules <i>Architecture:</i> Bottom-up abstraction hierarchy
Kashino 1993	Tokyo University	3	quite reliable <i>Sounds:</i> Flute, piano, trumpet (automatic adaptation to tone)	18 Perceptual rules, timbre models, tone memories, statistical chord transition dictionary <i>Architecture:</i> Blackboard, Bayesian probability networks
Martin 1996	MIT	4	quite reliable <i>Sounds:</i> Piano	33 Perceptual rules <i>Architecture:</i> Blackboard

- Klapuri [30], Cemgil [7] and others have applied Bayesian analysis to beat tracking and rhythm quantisation, as described in Section 2.6.
- Klapuri [30], Krumhansl [33] and others have used Bayesian probability analysis for key detection.
- Walmsley *et al.* [61] have investigated the use of Bayesian modelling to estimate harmonic model parameters, and the use of time-domain variation of model parameters to model the variation of the harmonic structure over time (as the sound goes through attack, sustain, decay). The general linear model, which is similar to the model in Equation 2.7, is used to describe the way note harmonics are superimposed in polyphonic music signals. This model is then analysed in a Bayesian framework.

In summary it can be said that Bayesian modelling is a powerful and commonly used way to introduce prior knowledge (chord transitions, harmonic models, etc.) into a system.

2.10 Comparison of Different Systems

In concluding the literature review, an overview of published transcription systems is given in Table 2.1.

From the table it can be seen that even two-and-a-half decades after the first published polyphonic transcription system, the degree of polyphony, the instruments and ranges of transcription systems are still very restricted.

Chapter 3

Basic Acoustics & Theory of Music

Music is an expression of our dreams, fears and hopes that dates back virtually to the dawn of our species. Music as an art is a human creation and developed out of our ancestor's experience of what constitutes "pleasant" and "unpleasant" sounds and combinations of sounds, and their desire to mimic these sounds from their surroundings [13]. Music exploits the characteristics of human hearing and perception and thus there are certain features common to all forms of music throughout the ages and in all cultures. However, as with all artforms, whether visual or aural, music also has certain features which are culturally determined. This has led to a vast array of different forms of music with different rules and syntaxes.

An investigation of the basic properties of human audition, acoustics and music theory will attempt to define more precisely what is meant by "music". Such a study also serves to outline the properties of human audition which can be incorporated in processing models and to determine the features of music which can be exploited to make automatic music transcription possible.

3.1 An overview of human audition

3.1.1 The human ear

A schematic diagram of the human ear is given in Figure 3.1. The outer ear collects sound with the pinna. The sound is then conducted through the auditory canal, which acts as a pipe resonator that boosts hearing sensitivity in the range of 2000 to 5000 Hz. The outer ear terminates in the ear drum, the beginning of the middle ear. The ear drum changes the slight pressure variations of incoming sound waves into mechanical vibrations with the help of the ossicles: three small bones shaped like a hammer, an anvil and a stirrup respectively.

The stirrup vibrates against the oval window of the cochlea in the inner ear. "The spiral

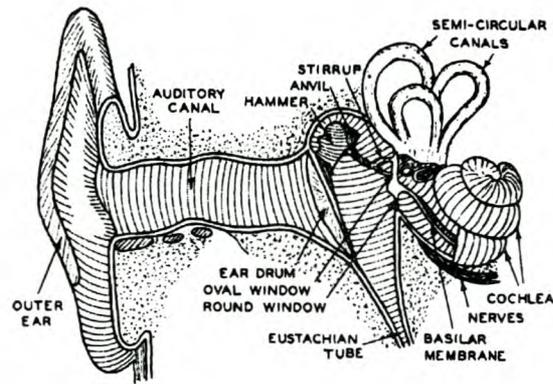


Figure 3.1: *The human ear (taken from [45, Fig 7.1])*

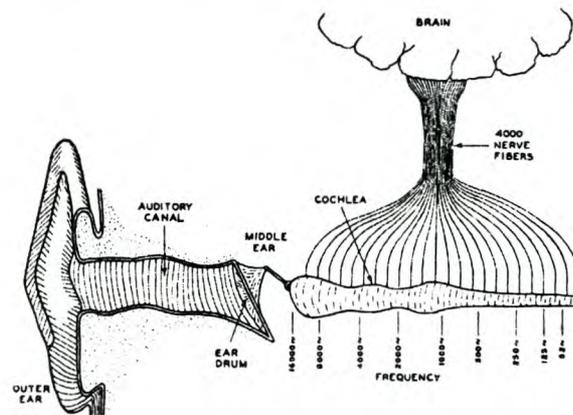


Figure 3.2: *Schematic view of the human hearing mechanism (taken from [45, Fig 7.3])*

cochlea, a masterpiece of minituarisation, contains all the mechanisms for transforming pressure variations into properly coded neural impulses” [53, p. 63]. When the cochlea is uncoiled (as in Figure 3.2), it appears as a tapered cylinder divided into two sections by the basilar membrane which runs down the length of the cochlea. The cochlea is filled with fluids which transmit pressure waves through its length when the stirrup vibrates against the oval window. The fluids in turn induces ripples in the basilar membrane. High tones cause the greatest ripples near the oval window where the basilar membrane is narrow and stiff, whilst low tones create the ripples with the largest amplitude where the membrane is slack at the far end. The mechanical vibrations are transformed into electrical neural impulses through the hair cells of the organ of Corti. When the membrane vibrates, the hairs are bent, which causes neurons leading to the brain to fire, depending on the intensity and frequency of the sound.

The great 19th century scientist Helmholtz studied the hearing mechanism and is considered one of the founding fathers of psychoacoustics. Rossing summarises Helmholtz’s conclusions as follows:

Helmholtz envisioned the fibers of the basilar membrane as selective resonators tuned, like the strings of a piano, to different frequencies. Thus a complex sound would be analysed into its various components by selectively exciting fibers tuned to the frequency of one of the components. [53, p. 66]

This is the essence of the “place theory” of hearing. Although modern researchers have determined that hearing likely also entails an operation akin to autocorrelation for finer pitch resolution of especially lower tones (the so-called “periodicity theory”), Helmholtz’s theories are nonetheless useful in getting a handle on the basic mechanism whereby humans perceive sound, namely that the human ear acts like a filter bank with logarithmically spaced centre frequencies.

3.1.2 Auditory streaming

The human ear, a marvel of biology in itself, is not sufficient on its own to account for all the amazing abilities of humans to track sounds even in noisy environments. Albert Bregman [3] systematised the study of human auditory perception. His main thesis is that human auditory perception is based on the Gestalt law of common fate, namely that

When different partials in the spectrum undergo the same change at the same time, they are bound together into a common perceptual unit and segregated from partials whose time-varying behavior is different. This principle applies both to changes in intensity and changes in frequency. [3, p. 394]

He gives four auditory cues based on common fate which the human auditory perception system uses for stream separation:

1. *Common onset*: If a number of frequencies start at exactly the same time, they are likely to originate from the same source.
2. *Harmonicity*: Frequencies that are part of a harmonic series are more likely to be grouped into a single auditory stream than harmonically unrelated frequencies.
3. *Common frequency variation*: Partial s whose frequencies vary at approximately the same rate are likely to belong to the same auditory stream.
4. *Common amplitude variation*: Partial s whose amplitudes vary at approximately the same rate are likely to belong to the same auditory stream.

The above cues can be used to group related partials into auditory streams, along with other cues such as *common spatial allocation* which states that sounds coming from the same spatial location are likely to be perceived as originating from the same source. However, Bregman also describes the mechanism whereby existing streams are traced:

3.2.1 Pitch

Pitch vs. Frequency

Although the term *pitch* is often used interchangeably with the term *frequency*, the two are not, in fact, equivalent. The former is a subjective quantity describing the “position of a sound in a musical scale” [14, p. 35], the latter is an objective value describing the number of oscillations per second of a sonorous body.

Though pitch is often related to the fundamental frequency of a sound, the apparent pitch f_{pitch} of a sound is sometimes linked to the difference tone produced by partials of frequency f_a and f_b :

$$f_{pitch} = |f_b - f_a| \tag{3.1}$$

such that a sound containing frequencies of 700, 800, 900 and 1000 Hz will typically will be perceived to have a pitch of 100 Hz. Moreover, it seems that the ear can pick out a series of nearly harmonic partials and determines the pitch to be the largest near-common factor of the series [53, p. 110]. For example, a sound with component tones of 1040, 1240 and 1440 is commonly perceived to have a pitch of 207 Hz because $1040/5 = 208$, $1240/6 \approx 207$, and $1440/7 \approx 206$, even though the difference between the partials is 200 Hz. Similarly, a complex tone with frequencies of 300, 500, 700 and 900 Hz will typically be perceived to have a pitch of 100 Hz [24, p. 58].

Pitch Duration

For a tone to produce a definite pitch, it has to be of a certain minimum duration that varies with frequency, though early experiments suggested that pitch develops after two cycles of the sound (shown as the dashed line in Figure 3.4). Modern research has shown that tones with a duration less than the solid line in Figure 3.4 are perceived as “clicks”.

Pitch Perception and Loudness

Pitch perception of pure tones (sinusoidal tones without overtones) of fixed frequencies varies somewhat according to the sound level: in some experiments the apparent pitch deviation was found to be up 1.3 semitones when the sound level was raised by 40 dB [14, p. 48], though modern research has shown that this effect is smaller than previously assumed [53, p. 108]. Fortunately, musical sounds are generally rich in overtones, and in such sounds the apparent pitch is largely unaffected by the sound level [14, p. 48], [53, pp. 108-109]. Because of the fact that apparent pitch and fundamental frequency so closely agree in musical sounds, this thesis will follow the common practice of using the terms *pitch* and *fundamental frequency* interchangeably.

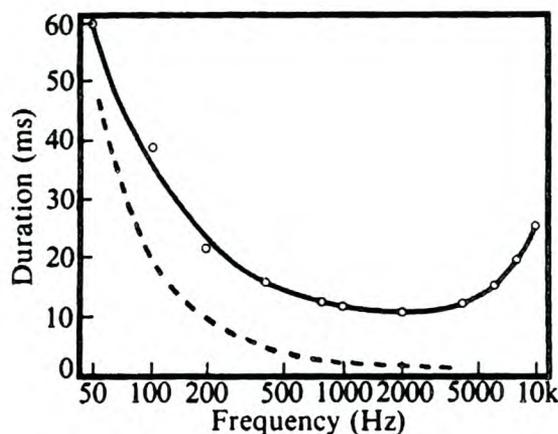


Figure 3.4: *Minimum durations for pitch sensation (taken from [53, Fig 7.8])*

Pitch Standard

Pitch perception is generally relative, meaning that most people can (given some training) hear and classify intervals between two tones. Very few people (< 0.01% of the population) are blessed with absolute pitch (“perfect pitch”) which allows them to recognise and define the pitch of a tone without the use of a reference tone. To standardise tuning a fixed pitch reference tone is provided according to which instruments are commonly tuned. Concert or standard pitch is defined as:

$$f_{A4} = 440 \text{ Hz} \quad (3.2)$$

Pitch Discrimination

Pitch discrimination, taken to be the smallest difference in pitch that humans can recognise, is dependent on frequency. Below 60 Hz, this just noticeable difference is nearly a semitone¹. In the frequency band of 500 and 4000 Hz where the ear is most sensitive, changes in frequency of around 0.3% or 5 cents² can be discerned. Pitch discrimination is also dependent to some extent on training: Culver mentions that skilful piano tuners can recognise the difference between a just-tempered and an equal-tempered³ fifth, implying a pitch discrimination of 2 cents [14, p. 47]!

¹Most people that I queried about this, myself included, cannot distinguish between the lowest two notes of the piano.

²Cent is a logarithmic unit of measurement for pitch, and is defined in Section 3.3.3.

³Temperament is discussed in Section 3.3.3.

3.2.2 Timbre

Harmonics occur at integer multiples of a fundamental frequency f_0 :

$$f_h = hf_0 \quad (3.3)$$

In practice, many musical instruments have “harmonics” (overtones) that occur at values that are approximately (but not exactly) integer multiples of the fundamental frequency. Thus the word *partial* has come to apply to signify the fundamental or one of its (not necessarily perfectly harmonic) overtones.

The timbre of a musical sound is chiefly determined by the number, intensity and distribution of partials that enter into its composition [14, p. 61]. However, the timbre of many instruments is also dependent upon the transient attack and thus the time envelope of the sound. In general, the upper partials are relatively strong during the attack, but due to damping decrease in amplitude before the steady phase of the tone.

As an example of a real musical instrument and its timbre properties, the piano will be briefly examined. Pianos produce their sound when strings are set into vibration by a “hammer”. An ideal string would vibrate in a series of modes that are exact harmonics of the fundamental. Actual strings have some stiffness, which creates a restoring force in addition to the tension, which slightly raises the frequency of all the modes. This restoring force is greater in the case of higher harmonics due to the greater number of bends in the string. The modes are thus spread slightly apart in frequency, and the partials are therefore no longer exact harmonics of the fundamental.

This relationship can be written as (cf. [53, p. 264]):

$$f_h = hf_0(1 + Ah^2) \quad (3.4)$$

where A is given for solid wires without wrapping by the equation:

$$A = \frac{\pi^3 r^4 Y}{8Tl^2} \quad (3.5)$$

where r is the radius of the string, Y is Young’s modulus, T is the tension and l is the length of the string. The ramification of this is that the tuning of pianos is “stretched” to allow the slightly stretched upper partials of a lower note to coincide with the lower partials of a simultaneously sounded higher note and thus reduce dissonance. Although piano tuning deviates very little from equal temperament in the middle registers, low notes are typically down-tuned by as much as 60 cents. Conversely, extremely high notes are typically tuned much higher than the true equal-tempered values.

Typical piano spectra are given in Figure 3.5. It can be seen that different notes have different spectral characteristics. It should also be mentioned that spectral characteristics

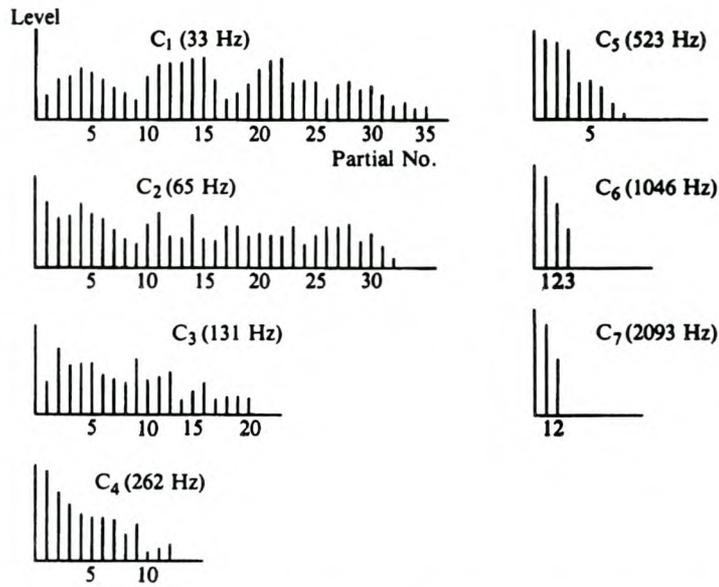


Figure 3.5: *Piano spectra for various notes (taken from [53, Fig 14.5])*

are a function of amplitude, and thus a loud note will typically have a different spectrum compared to a soft note at the same pitch. The dependence of timbre on pitch and loudness complicates accurate instrument modelling significantly.

3.2.3 Loudness

Sound levels

Sound waves are, in effect, minuscule variations in atmospheric pressure to which our hearing responds, as described in Section 3.1. Although the exact values differ, the pressure amplitude at the threshold of pain is approximately 10^6 greater than at the threshold of hearing. Because of this tremendous range in values, sound pressure levels L_p are measured on a logarithmic scale, the decibel (dB) scale:

$$L_p = 20 \log p/p_0 \quad (3.6)$$

where p is the sound pressure and $p_0 = 20 \mu\text{Pa}$ is the sound pressure reference for a sound at the threshold of hearing. This places the threshold of pain at 120 dB.

Loudness and musical dynamics

The sensitivity of the ear varies with frequency, being relatively insensitive to very low-pitched (< 50 Hz) and very high-pitched (> 15 kHz) sounds, with two very sensitive regions at 3500-4000 Hz and 13 kHz which correspond with the resonance frequencies of

the outer ear canal. A scale in units of *phon* is used to measure loudness levels, and graphs of equal loudness curves can be found in many standard reference works like Rossing [53] to convert decibels to phons. Another unit of measurement is used to indicate the subjective loudness of tones: the *son*, with conversion curves provided again in Rossing and others.

Both of these scales are however flawed when applied to music perception: firstly, they are *subjective* scales that vary considerably from person to person, and secondly, they are generally only given for loudness perception of *pure* tones. Calculating values for the loudness of complex sounds is an involved process, though the formula

$$L_S = L_m + 0.3 \sum_i L_i \quad (3.7)$$

can be used to calculate the loudness L_S in sones of the complex sound S , using loudness indexes for the perceived loudness of the sound content in 10 standard octave bands. The loudness indexes for these bands are given by ISO Recommendation No. 532 [53, pp. 87-88]. In the above formula, L_m is the greatest loudness index, and $\sum L_i$ is the sum of the remaining indexes. Equation 3.7 suggests that the perceived loudness of a complex sound is determined chiefly by the loudest sound in a complex mixture.

Variations in loudness (called *musical dynamics*) are a major tool for expressive musical performance: loud sections are more appropriate for triumphant climaxes or outbursts of musical “anger” whilst soft sections may better convey a sense of intimacy and introspection. Musical dynamics are often suggested in the score, and are generally indicated in six levels from *pp* (pianissimo = very soft) to *ff* (fortissimo = very loud), or eight levels from *ppp* to *fff*. Studies have been conducted into the maximum dynamic ranges which instrumentalists use and it was found that the average maximum dynamic range is only around 10 dB (with slight variations for different instruments) for a given note played loudly and softly [53, p. 90]. This suggests that most instrumentalists would have a hard time producing six (or even eight) distinguishable levels of loudness. Thus it seems that the dynamic indications for a musical passage are much more an indication of a sentiment that has to be evoked rather than a required actual sound level. It should also be noted that although the dynamic range on a given note is fairly small, many instruments are capable of producing much louder tones at the top of their range than at the bottom. A fortissimo on a French horn is found to be nearly 30 dB greater at C_5 than at C_2 , according to [53].

Implications for automatic music transcription

The above discussion of sound and loudness levels has some implications for automatic music transcription:

- It may prove non-trivial to calculate the perceived loudness of notes and to convert these values to musical dynamic indications.
- Because the dynamic range of music covers approximately 40 dB [53, p. 89], a musical performance may contain a mixture of sounds that differ in amplitude by a factor of 100 or more. It is thus very possible that louder sounds in the mixture will mask softer sounds during pitch extraction.

A complete automatic transcription system should address these issues.

3.2.4 Superposition

In general, the superposition of two or more sounds is a simple addition of their waveforms. Because the spectrum is calculated through a linear transformation of the time waveform, the superposition spectrum can also be obtained by a simple summation of the individual complex spectra.

For two pure tones of identical frequency f but possibly different amplitudes (A_1 and A_2) and different phases (θ_1 and θ_2), the superposition will be a pure tone of frequency f with an amplitude A_S in the range of $|A_1 - A_2| \leq A_S \leq A_1 + A_2$ depending on $\Delta\theta = |\theta_1 - \theta_2|$.

When two pure tones of different frequencies f_1 and $f_2 = f_1 + \Delta f$ are superimposed, the resulting combination tone S will be given by:

$$S = A_1 \sin(2\pi f_1 t + \theta_1) + A_2 \sin(2\pi f_2 t + \theta_2) \quad (3.8)$$

$$= A_1 \sin(2\pi f_1 t + \theta_1) + A_1 \sin(2\pi f_2 t + \theta_2) + (A_2 - A_1) \sin(2\pi f_2 t + \theta_2) \quad (3.9)$$

$$= 2A_1 \sin \left[2\pi \frac{(f_1 + f_2)}{2} t + \frac{(\theta_1 + \theta_2)}{2} \right] \cos \left[2\pi \frac{(f_1 - f_2)}{2} t + \frac{(\theta_1 - \theta_2)}{2} \right] + (A_2 - A_1) \sin(2\pi f_2 t + \theta_2) \quad (3.10)$$

where (3.10) was obtained from (3.9) by applying the trigonometric identity:

$$\sin A + \sin B = 2 \sin \frac{1}{2}(A + B) \cos \frac{1}{2}(A - B) \quad (3.11)$$

From (3.10) it can be seen that if the component tones have similar amplitudes (such that the second term is negligible) and Δf is small, the superposition will be a tone at the average frequency $f = \frac{1}{2}(f_1 + f_2)$ (from the sin factor) and an envelope that is amplitude modulated with the difference frequency Δf (from the cos factor, taking note that the envelope taken from absolute amplitude peak to absolute amplitude peak has *half* the period of the modulating cos). Since the ear has a finite pitch resolution, even sounds of dissimilar amplitudes will be perceived to be pitched at the “average” frequency if Δf is small.

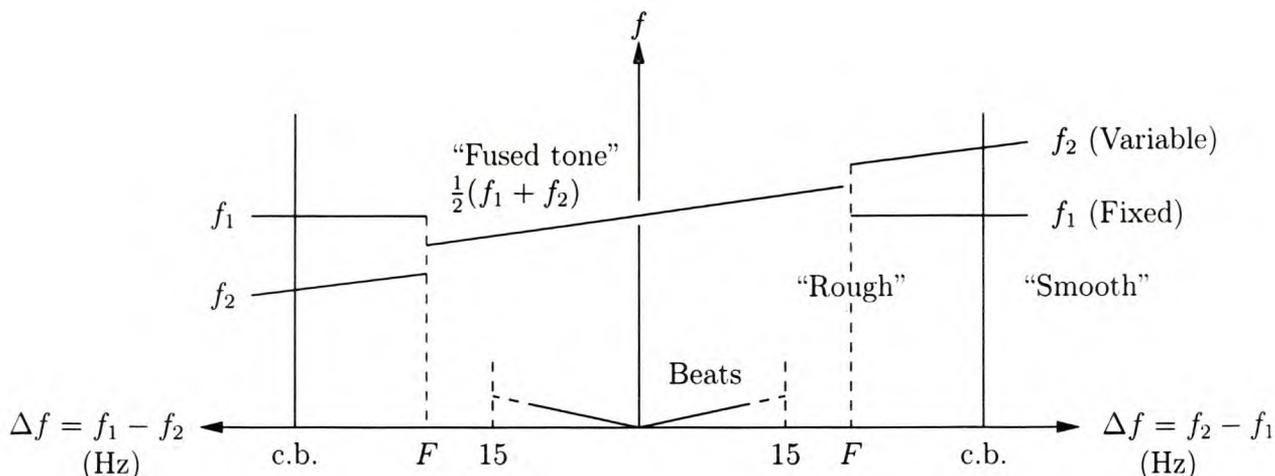


Figure 3.6: *Superposition of two tones (based on [53, p. 132])*

If Δf is less than around 10 Hz, the relatively slow change in the envelope's amplitude is perceived as audible (and uncomfortable!) beats. Although these beats are highly undesirable under normal performance conditions, they are used to tune instruments such as the piano by reducing the number of beats that result from the interference of a note and a reference tone. As Δf increases above 15 Hz, the beat sensation disappears and is replaced by a sensation of auditory roughness. If Δf is increased above some frequency F the perceived tone at the average frequency is replaced by two tones at the frequencies f_1 and f_2 , though the sensation of roughness remains until Δf exceeds some critical value related to the critical bands of hearing. This process is depicted in Figure 3.6.

The “roughness” due to the interference of two tones described above gives rise to the phenomena of consonance and dissonance. Helmholtz described the origin of these phenomena

[...] by referring to Ohm's acoustical law, which stated that the ear performs a spectral (Fourier) analysis of sound, separating a complex sound into its various partials. Helmholtz concluded that dissonance occurs when partials of the two tones produce 30-40 beats per second. The more the partials of one tone coincide with the partials of the other, the less chance of beats in this range that produce roughness (dissonance). This explains why simple ratios define the most consonant intervals. [53, p. 138]

More recent studies have shown that the frequency differences Δf which produce auditory “roughness” differ with frequency, leading to the critical bands of hearing mentioned before, though Helmholtz still summarised the causes of consonance and dissonance very aptly. This discussion of consonance and dissonance serves as background for the theory of diatonic scales in Section 3.3.2.

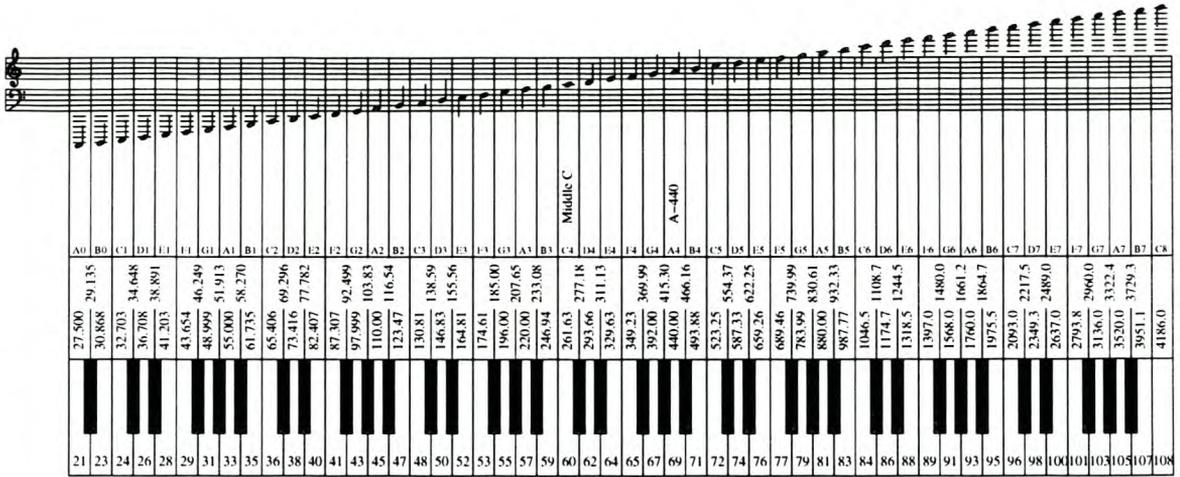


Figure 3.7: Piano keyboard with notes (MIDI), pitch (Hz), octave registers and staff notation

3.3 Basic music theory

Given the vast corpus of world music, the scope of music under consideration needs to be narrowed down to reduce the complexity of the transcription problem. Thus for the purpose of this thesis, we shall limit our theory to that of Western music based on diatonic scales. It is hoped that future work will be extended to include other forms of music, such as various traditional African musical expressions.

3.3.1 Notation

Music is a “language”, with rules and vocabulary of its own. As with any language, a standard form of notation is crucial for written transmission.

The Keyboard and Octave Registers

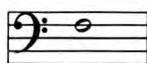
Pitches are named according to the first seven letters of the alphabet, A, B, C, D, E, F and G [32, p. 3]. This pitch alphabet is related to the piano keyboard using C as a reference. The C nearest to the middle of the keyboard is called *middle C* or C₄ because it falls into the fourth octave register. Every note that falls within the same octave as a given C (i.e. all 12 notes from a C up to B) is said to lie in the same octave register. The C an octave above middle C is thus named C₅, the one below C₃ etc. Refer to Figure 3.7 for a visual representation of this convention (which will be used throughout the rest of this thesis). A more detailed version of this figure is given in Appendix A.

It should be noted that the term *pitch class* is used “to group all pitches that have an identical sound or that are identical except for the octave or octaves that separate them”

[32, p. 52]. Under equal-tempered tuning, the notes $B\sharp$, C and $D\flat$ belong to the same pitch class, as do C_4 , C_5 , C_3 etc.

Notation on the Staff

Musical pitch is distributed exponentially in frequency (as shall be seen presently in Section 3.3.3). For “plotting” pitch, a form of semi-logarithmic plot is used: the musical *staff*. The logarithmic “ y axis” indicates pitch and the “ x axis” indicates time. A modern staff consists of five lines and four spaces, with ledger lines used to extent the staff up and down if necessary. To indicate the pitch reference of the lines, two clefs are commonly used: the G (or treble) clef, whose “curl” indicates the position of G_4 , and the F (or bass) clef, which indicates the position of F_3 .



F3



G4

The notes on the staffs without any accidentals indicate the notes of the C major scale (the white notes on the piano). Accidentals are used to modify the pitch as it is displayed. A sharp (\sharp) indicates that the pitch of the note is to be raised a semitone, a flat (\flat) indicates that the displayed pitch is to be lowered a semitone, a natural (\natural) restores the pitch of the note (i.e. cancels any previous accidentals), a double sharp (\times) raises the displayed pitch by two semitones, and a double flat ($\flat\flat$) lowers the pitch by two semitones.

Except for the C major and natural a^4 minor scales, all scales contain notes corresponding to the black notes on the piano. It would be tedious to indicate these pitches with accidentals every time they occur. Thus a key signature is used at the beginning of every staff to indicate which pitches should be raised/lowered throughout the piece. For example, following the note intervals given in Section 3.3.3, D major is defined as D , E , $F\sharp$, G , A , B and $C\sharp$. The key signature is said to contain two sharps, $F\sharp$ and $C\sharp$. The order in which sharps are added to the key signature is as follows: $F\sharp$, $C\sharp$, $G\sharp$, $D\sharp$, $A\sharp$, $E\sharp$ and finally $B\sharp$. The line “*Father Charles Goes Down And Ends Battle*” is frequently used to remember the increasing order of sharps. Flats are added in the reverse order, and thus the line “*Battle Ends And Down Goes Charles’ Father*” can be used as a mnemonic. To determine the number of accidentals in the key signature for different scales, the circle of fifths is often used. It gives the key signatures for both major scales and their related minor scales, as shown in Figure 3.8.

⁴The key of major scales is generally capitalised, whilst the key of minor scales is usually given in lower-case.

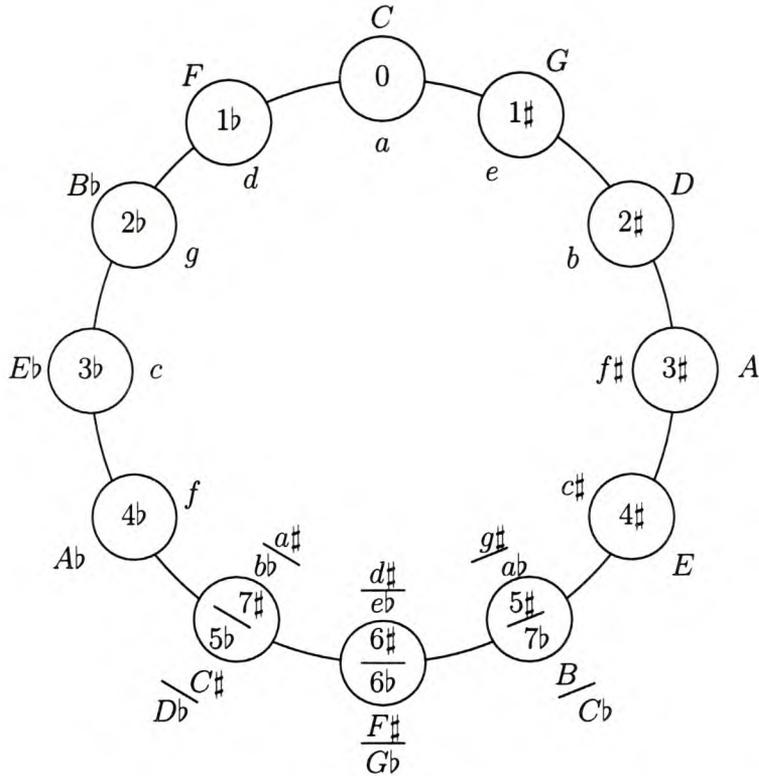


Figure 3.8: Circle of fifths for key signatures (based on [32, p. 15])

Notation on the Piano Roll

There are times when notation on the staff is impractical, especially when plotting segments for which only the pitch (in cents/semitones) and duration (in seconds) is known, but not the tempo, time signature or key signature. This situation is common when inputting notes on MIDI keyboards, and also during the early stages of music transcription before the “musical context” of the note candidates is determined. In this case, pitch is plotted on a semitone grid placed against a vertical piano keyboard (simultaneously the Y axis and the axis legend), with time being the horizontal dimension. An example of a plot on a piano roll is Figure G.2 in the experimental investigation.

3.3.2 Diatonic Scales

As mentioned in Section 3.2.1, many humans can recognise specific ratios of frequencies. Furthermore, most humans can recognise that certain combinations of tones induce a pleasing auditory effect, whilst other combinations produce an unpleasant effect (keeping in mind that “pleasing” and “unpleasant” are to some extent determined by the cultural milieu). In general, musical intervals are given by:

$$f_2/f_1 = m/n \quad (3.12)$$

Table 3.1: *Pleasant Note Intervals (based on [24, p. 80])*

<i>Interval Name</i>	<i>Ratio</i>	<i>Example</i>
Unison	1:1	$C_4 - C_4$
Octave	1:2	$C_4 - C_5$
Perfect Fifth	2:3	$C - G$
Perfect Fourth	3:4	$C - F$
Major Sixth	3:5	$C - A$
Major Third	4:5	$C - E$
Minor Sixth	5:8	$C - A\flat$
Minor Third	5:6	$C - E\flat$

Table 3.2: *Construction of the C Major Scale*

C	D	E	F	G	A	B	C'	D'
4	5	6				
			4	5	6	
				4	5	6

where m and n are generally small integers for “pleasing” ratios. Values for such pleasant diads are given in Table 3.1.

These intervals are considered “pleasant” or consonant because the dissonant combinations between harmonics of the two tones are minimal.

Similarly, certain musical triads (three notes) are also particularly pleasing to the ear. Remarkably, the most pleasing of these triads (called a major triad) consists of tones whose frequencies bear the ratio 4:5:6. The C major scale is constructed from the three major triads $C - E - G$, $F - A - C$ and $G - B - D$ on the first, fourth and fifth notes of the scale respectively as shown in Table 3.2.

From Table 3.2 it becomes simple to calculate the ratios relative to C of each note of the scale:

$$\begin{aligned}
 E &= 5/4C \\
 G &= 6/4C = 3/2C \\
 B &= 5/4G = 5/4 \times 3/2C = 15/8C \\
 D' &= 6/4G = 3/2 \times 3/2C = 9/4C \\
 D &= D'/2 = 9/8C \\
 A &= 5/6C' = 5/6 \times 2C = 5/3C \\
 F &= 4/6C' = 2/3 \times 2C = 4/3C
 \end{aligned}
 \tag{3.13}$$

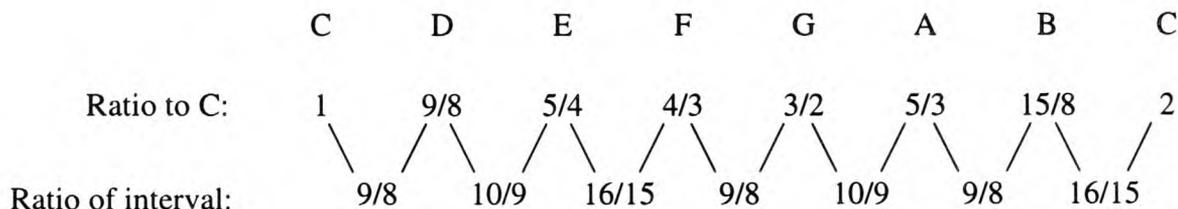


Figure 3.9: *Frequency ratios in the diatonic major scale (based on [53, Fig 9.1])*

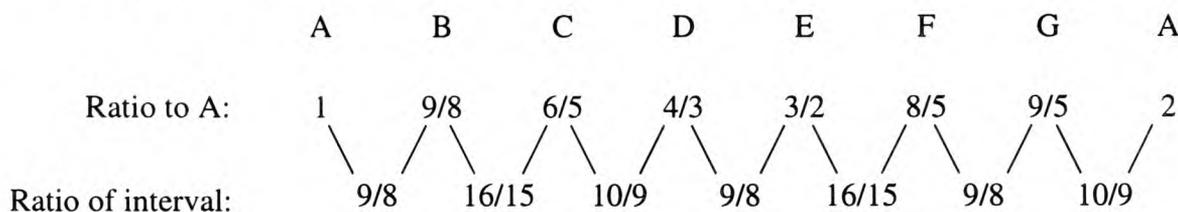


Figure 3.10: *Frequency ratios in the diatonic minor scale (based on [14, p. 79])*

The above results of the ratios of each note relative to C, as well as the ratios between successive notes are demonstrated in Figure 3.9.

In addition to scales constructed from the major triad, scales can be constructed with the minor triad with ratios of 10:12:15. Following a similar procedure to the one described above for major scales, the minor scale can be constructed from triads on the first, fourth and fifth notes of the scale respectively. The resulting minor scale is shown in Figure 3.10.

It is interesting to note that only three ratios between notes are involved, namely the major whole tone ($\frac{9}{8} = 1.125$), the minor whole tone ($\frac{10}{9} \approx 1.111$) and the semitone ($\frac{16}{15} \approx 1.067$). The semitone interval is slightly greater than half a major whole tone interval ($\frac{16}{15} \times \frac{16}{15} \approx 1.138$).

3.3.3 Temperament

Equal Temperament

The discussion in the foregoing section describes the diatonic scale, with intervals of just intonation. In practice, just intonation proves to be impractical because it does not allow for key changes without re-tuning the instrument. For example, in the C major scale as shown in Figure 3.9, the interval between D and E (the second of the scale) is a minor whole tone or $\frac{10}{9}$. When the scale is transposed to D, the interval between D and E (the first of the scale) is a major whole tone or $\frac{9}{8}$. If provisions were made for all possible key changes, instruments with fixed tones (such as the piano or the oboe) would require at least 72 notes to the octave [14, p. 80].

Thus equal temperament was devised whereby the octave is divided into 12 notes

Table 3.3: *Equal tempered scales*

<i>Scale type</i>	<i>Note Intervals in Semitones</i>
major	2 2 1 2 2 2 1
natural minor	2 1 2 2 1 2 2
harmonic minor	2 1 2 2 1 3 1
melodic minor ascending	2 1 2 2 2 2 1
melodic minor descending	2 1 2 2 1 2 2

(semitones), separated in frequency by a factor of $2^{1/12} \approx 1.05946$. Whole tones are then simply spaced two semitones apart, a factor of $2^{2/12} \approx 1.12246$. Whilst this compromise reduces the complexity of scale construction considerably, it has the unfortunate side effect that most notes in any given scale are now slightly false (as shown in Table A.1 in Appendix A).

Equal temperament has the advantage of greatly simplifying music theoretical calculations. The pitch frequency f_M in Hz of any given note M (numbered according to the MIDI standard, whereby middle C is numbered 60 and $A_4 = 440$ Hz is numbered 69) can be calculated as:

$$f_M = 440 \times 2^{\frac{M-69}{12}} \quad (3.14)$$

and conversely, the note can be calculated from the pitch as:

$$M = 12 \log_2 \frac{f_M}{440} + 69 \quad (3.15)$$

Since it becomes tedious to compare different tones using ratios due to their exponential spacing, intervals are generally specified in cents. A semitone is divided into 100 cents spaced a factor $2^{1/1200}$ apart in frequency. An interval specified in cents has the same musical meaning across all octaves because *cent linearises pitch*.

Although equal temperament will be assumed by our transcription system to simplify the algorithms, it should be noted from Table A.1 that all pitches of the just-tempered scale round to the same nearest 100 cents as the equal-tempered values.

Equal-tempered scales

Although the major and minor scales in use today were derived as discussed in Section 3.3.2, equal temperament allows for the easy definition of scales in any key by simply making use of the note intervals given in Table 3.3. The chromatic scale, composed of all of the twelve chromatic semitones, is omitted from the list.

Table 3.4: *The relationship between the harmonic series and pitch*

n	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
f_n	f_0	$2f_0$	$3f_0$	$4f_0$	$5f_0$	$6f_0$	$7f_0$	$8f_0$	$9f_0$	$10f_0$	$11f_0$	$12f_0$	$13f_0$	$14f_0$	$15f_0$	$16f_0$
Δc	0	1200	1902	2400	2786	3102	3369	3600	3804	3986	4151	4302	4441	4569	4688	4800
ΔM	0	12	19 ⁺	24	28 ⁻	31 ⁺	34 ⁻	36	38 ⁺	40 ⁻	42 ⁻	43 ⁺	44 ⁺	46 ⁻	47 ⁻	48

Note that the minor scale can be arrived at by right-rotating the intervals of the major scale by two positions. The consequence of this is that every minor scale has a related major scale with which it shares a key signature. This fact will be exploited by the transcription system to extract the best key signature from transcribed data.

The Harmonic Series and Equal Temperament

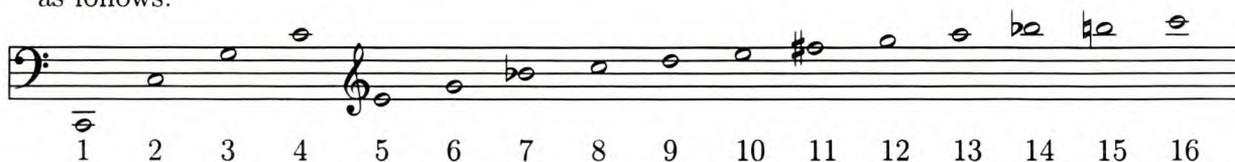
The notes of the equal-tempered scales constitute a geometric progression, whilst the harmonic series is an arithmetic progression. In processing musical spectra, it is often useful to express the frequency f_h of the h -th harmonic (as given by Equation 3.3) of a certain note frequency f_0 in terms of an interval in cents Δc or semitones ΔM , as follows:

$$\begin{aligned}\Delta M &= 12 \log_2 \frac{f_h}{f_0} = 12 \log_2 \frac{hf_0}{f_0} \\ &= 12 \log_2 h\end{aligned}\tag{3.16}$$

$$\Delta c = 1200 \log_2 h\tag{3.17}$$

These equations yield the results in Table 3.4, where ⁺ indicates that the harmonic is “sharper” than the corresponding note at that position and ⁻ indicates that the note is “flatter”.

The relationship between C_2 and its harmonics can be represented in musical notation as follows:



3.3.4 Meter and Rhythm

Music has a horizontal and a vertical aspect, the former being rhythm, the latter being harmony. Rhythm is related to note duration, and thus a set of symbols is used to indicate the relative durations of notes, so that each symbol represents twice the duration of the next shorter symbol. Duration symbols for commonly used notes are given in Table A.2. A dot extends the length of a note by half its value, so that $\text{♩} = \text{♩} + \text{♪}$ and $\text{♩} = \text{♩} + \text{♪} + \text{♪}$.

The *beat* is the basic pulse of a passage. The rate at which the beat occurs (the “foot-tapping rate”) is the *tempo* of the piece. Tempo is specified either qualitatively

Table 3.5: *Meter types (based on [32, p. 28])*

<i>Grouping</i>	<i>Meter type</i>	<i>Metric accent pattern</i>
Two-beat measure	Duple	Strong-weak
Three-beat measure	Triple	Strong-weak-weak
Four-beat measure	Quadruple	Strong-weak-less strong-weak

with Italian words like *Allegro* (=fast) or quantitatively by a metronomic indication like $\downarrow = 72$ which specifies the number of beats per minute (72 in this case).

Beats can be generally grouped into patterns that remain consistent throughout a passage; such patterns are the *meter*. Groups of two, three or four beats are most common, although other meters occur. The groups of beats are called *measures* and in musical notation, the end of a measure is indicated with a vertical line through the staff called a *bar line*. The most common classical meter types are summarised in Table 3.5, along with the typical metric accent (stress) patterns for each.

In most music, the beat is divided into shorter durations. Beats that are generally divided in two equal parts are called *simple beats*, whilst beats that are generally divided into three equal parts are called *compound beats*. The smallest note duration that is found more than incidentally is called the *tatum*. The division of the beat and measure is summarised by the time signature:

A *time signature* is a symbol that tells the performer how many beats will occur in each measure, what note value will represent the beat, and whether the beat is simple or compound. [32, p. 31]

Typical time signatures are given in Table A.3.

Note that many time signatures are functionally equivalent. For example, a $\frac{6}{8}$ piece can be notated as $\frac{6}{4}$ with all note duration symbols scaled by a factor 2. This is just one of numerous ambiguities in the identification and notation of rhythm; another ambiguity comes from the fact that the identification of meter is often a matter of interpretation of stress patterns [32, p. 28]. The ambiguity of meter and rhythm implies that even sophisticated automatic transcription systems with complex metric analysing components may not be able to reproduce the original rhythmic notation without human intervention.

3.3.5 Polyphony defined

For the automatic transcription problem, the terms *monophony* and *polyphony* are often contrasted against each other to designate two categories of signals. It seems that software

engineers and musicians at times differ in their definitions of these terms. *The New Grove Dictionary of Music and Musicians* defines the terms as follows:

monophony Music for a single voice or part. In many non-Western cultures it may have improvised or drone accompaniment. [18]

homophony Form of polyphony with rhythmic similarity in a number of parts or in which all melodic parts move together at more or less the same pace. Many instances of choral church music are examples of homophony. [23]

heterophony The simultaneous sounding of a melody and variations of it. [12]

polyphony Music in more than one part, music in many parts, or the style in which all or several of the musical parts move to some extent independently. [19]

From the above definitions it can be seen that polyphony can refer to several distinct categories of music, whilst monophony may include simple accompaniment. For the purposes of this thesis, the terms shall be defined as follows:

monophony Music for a single voice or part, without any form of accompaniment (i.e. musical waveforms containing only one distinct sound).

polyphony Music for more than one voice or part (i.e. musical waveforms that are a mixture of more than one distinct sound).

This is in line with the usage of the terms by researchers in the automatic music transcription field.

3.3.6 Harmony

Harmony is defined by Kostka [32, p. ix] as follows:

Harmony is the sound that results when two or more pitch classes are performed simultaneously. It is the vertical aspect of music, produced by the combination of the components of the horizontal aspect.

Furthermore, *tonal harmony* is of special significance, as the Western music composed during the period from 1650 to 1900 made almost exclusive use of tonal harmony, although it developed much earlier than that and is still employed today in many genres of music (of which popular music is perhaps the most significant, at least in volume). Tonal harmony can be outlined as follows [32, p. xi]:

1. Tonal harmony makes use of a *tonal centre*, a key pitch class that provides a centre of gravity.
2. It makes almost exclusive use of major and minor scales.

3. Chords are primarily tertian in structure (i.e. are built from intervals of thirds).
4. Chords built on various scale degrees relate to each other and to the tonal centre in fairly complex ways. However, each chord has a standard role (function) within a key, and thus the term *functional harmony* is often used to refer to this kind of music.

Music based on tonal harmony is well-suited for the development of automatic music transcription systems because of the well-defined structure of chords as well as their relation to each other. Such systems can make use of the rules of tonal music as a probabilistic knowledge source to identify tonal centres, keys, chords and harmonic progressions, and to enhance the accuracy of the transcription.

3.4 Problems to be solved for complete music transcription

Initially it may seem that the problem of automatic music transcription is the sheer impossible task of solving a mixture S of N superimposed sounds for the component sounds S_c from the single equation:

$$S = \sum_{c=1}^N S_c \quad (3.18)$$

Fortunately, this chapter outlined a number of characteristics of the human auditory system, musical sound and music theory which can be used to solve the transcription problem.

The complete transcription problem can be divided into the following sub-tasks:

- *Polyphonic pitch detection*: Perhaps the foremost and most crucial task in music transcription is to identify the component pitches in a mixture of simultaneous musical sounds. For full transcription, these sounds need not be restricted to voiced harmonic sounds but should be extended to include percussive sounds also.
- *Instrument recognition*: Hand-in-hand with the previous point, the instruments which generated each sound should be identified.
- *Rhythm and meter detection*: An important task in music transcription is identifying the tatum, beat and measure pulses.
- *Time signature and tempo recognition*: Based on the results of the rhythmic analysis, the time signature and tempo should be detected. It should be noted that the time signature and tempo can change from one segment of a piece to the next.

- *Key detection:* The key of a piece needs to be identified if one wants to conduct an analysis of harmony. It is also used to notate pitch in such a way that the meaning of each note becomes apparent. The key can also change several times during the course of a piece.
- *Identifying harmony and harmonic progression:* Notes that were sounded concurrently can be grouped into chords, each with a harmonic root and a chord type. The sequence in which chords are sounded (i.e. the harmonic progression) is generally well defined for certain types of music.
- *Identifying expressive performance features:* Ornamentation (like trills, mordents and glissandos) and other performance indications (such as staccato, sforzando and legato) need to be identified appropriately. Furthermore, the loudness of note sequences and the changes in musical dynamics (crescendos, decrescendos) need to be identified.
- *Voice and part segmentation:* Notes generated on a certain instrument need to be grouped into parts. Furthermore, for notes sounded on a certain instrument, each note has to be assigned meaningfully to a specific voice. After this step, the original signal should have been segmented into individual voices in such a way that no voice contains more than one simultaneous note.
- *Notation:* The results of the analysis should be notated according to some convention.

Solving all of the above tasks comprehensively and satisfactorily would involve a system that is comparable in conceptual complexity to a complete speech recognition engine. Following the investigative path chosen in this chapter, our transcription system is based on the theory of tonal, non-percussive, equal-tempered Western music which can be expressed in standard modern music notation.

The insights gleaned from the investigation of acoustics and music theory are crucially important in the development of the algorithms in the following chapters even though not all aspects of music as outlined above are covered by the current transcription system. However, the references in this chapter can serve as the basis for further research into those neglected areas.

Chapter 4

System Design

Armed with knowledge from the musical and engineering domains, the focus of the discussion shall now turn to the design of an automatic transcription system that fulfills some of the requirements of complete transcription, as set forth in Section 3.4. This chapter will outline our transcription system and describe its restrictions. Chapters 5 and 6 will develop the individual system components in greater depth.

4.1 System components

4.1.1 Overview

The transcription system takes digitised waveform files of musical performances as input. Generally, the input signals need to be *pre-processed* to ease further processing.

With the data in a processable format, a full transcription system then requires three major components for extracting “raw” information from the music waveform:

1. a component that determines beat, meter and rhythm,
2. a component that determines polyphonic pitch, and
3. a component that determines which instruments were used to generate the sounds.

There is a large degree of correlation between the components: pitch and harmony changes are generally aligned with the beat and meter whilst multi-pitch estimators will often make use of instrument models to enhance their accuracy. The results from these components are then combined and submitted to further processing to try to enhance the raw transcription by comparing the results with expectations about the signal, using knowledge sources such as music theory.

Each of the system components will now be described in greater detail in order to outline the significance of each component in the context of the transcription problem.

4.1.2 Preprocessor

Many audio signals that are suited for transcription are recorded in 16-bit stereo at a sampling frequency of 44.1 kHz. Although stereo signals may contain one of Bregman’s cues for auditory stream separation, namely *spatial allocation*, the current system does not make use of it¹. Thus it is desirable to downmix the signal to mono as follows:

$$x_{mono}[n] = \frac{x_{left}[n] + x_{right}[n]}{2} \quad \text{where } 0 \leq n \leq L - 1 \quad (4.1)$$

4.1.3 Extracting physical information from the waveform

A meaningful first step in processing the waveform is to perform a rhythmic analysis of the signal to detect the time instants when musical “events” occur. The most important event type that can be analysed is the *note onset*. The attacks of notes generally show up in the waveform envelope as amplitude spikes (for instruments with a strong attack) or show up as large positive changes in the power distribution of the signal in various frequency bands. *Note endings* are difficult to determine precisely for instruments like the piano whose sound decays naturally over time after the initial attack. On instruments with sustained sounds, note endings can be detected as large negative changes in the power envelopes in corresponding frequency bands.

If note onsets (and note endings of sustained notes) can be detected accurately, the signal can be assumed to be quasi-stationary in the intervals between these events. This has a number of advantages for multi-pitch analysis:

- The multi-pitch estimator can dynamically adapt the analysis frame size: in slow passages, longer frames can be chosen to trade time resolution for greater frequency resolution; in fast passages, shorter frames can be chosen to allow for finer time resolution.
- Labelled note events also allow for the alignment of the analysis frames to note borders so that the analysis is not “blurred” across note boundaries.
- It is computationally more efficient to perform multi-pitch analysis only when there is a significant change in the signal, instead of analysing the signal with a constant frame rate.

Pitch analysis consists of two components, pitch *estimation* and pitch *tracking*. The pitch estimator finds a list of component pitches for each analysis frame. The pitch

¹In fact, no complete published transcription system in literature caters for this auditory cue. Future research into ways of using the distribution of sounds in stereo space to aid in stream separation may well prove very useful in enhancing the accuracy of transcription systems.

tracker then tries to trace pitch components across consecutive frames to form notes with a definite onset and duration.

A third component of a transcription system that extracts “raw” information is an instrument detector. If the individual parts making up a multi-instrument score are to be distinguished, an instrument detector has to be implemented to determine which instrument generated each note. An instrument detector would need training data for each instrument to be detected, preferably at different pitches and intensity levels, from which it can construct timbre models using the spectrum and time envelope. Such a set of training data generally comes in the form of a sample bank. Instrument models extracted from instrument sample data can also be used in the development of more sophisticated algorithms to separate sounds with partially or completely overlapping spectra. The problem of overlapping spectra is discussed in the following chapter.

The combined results of these three components is ideally a set of “raw” notes with attributes that describe:

- *Timing*: the note onset and duration, in seconds or samples
- *Pitch*: the fundamental frequency, in Hz or cents
- *Timbre*: the instrument whose characteristics best match the sound
- *Loudness*: the relative intensity of the sound, in dB or another appropriate measure

These values describe the *physical* attributes of the sound. Being able to transcribe a complex piece accurately to such a “raw” score would already be a major accomplishment and would constitute a huge leap towards a complete transcription solution. In fact, the attributes listed above are sufficient to transcribe the waveform accurately to a MIDI representation. However, in order to generate a score in standard musical notation, the raw notes need to be placed in an appropriate musical context which gives shape to the structure of the music.

4.1.4 Integrating the musical information

Each of the four attributes of raw notes need to be transformed to values that have musical meaning. Most importantly, the pitch needs to be discretised to chromatic semitones, and note timings need to be discretised to musical note durations.

The notes also need to be grouped *vertically* to form chords and *horizontally* to emphasise their metric structure. Notes belonging to a particular instrument need to be grouped into a part for that instrument. Additionally, expressive performance characteristics such as dynamics and ornamentation can be identified to give texture to the score.

Key and time signatures need to be identified for the piece or segments of the piece. Notes and chords then need to be notated sensibly according to the key signature, and grouped correctly into measures according to the time signature.

Most of these steps are incredibly complex. This stems partly from the inherent complexity of music as a medium which encompasses many different musical styles with different underlying rules. Transcription is further complicated by the fact that performed music is an expressive and artistic rendition of the original score, and not a precise and rigorous execution of it. Thus there are two sources of errors in the transcription process:

- *Errors caused by flawed analysis:* At virtually every step in the transcription process, decisions need to be made as to what value best represents a particular aspect of the signal (be it pitch, rhythm, key signature, etc.). Inevitably, some incorrect decisions (possibly based on flawed assumptions) will be made, giving rise to transcription errors. These *machine* errors can be eliminated or minimised by increasing the scope of the knowledge sources which the system employs, as well as by implementing top-down processing structures to allow for decision-making based on knowledge extracted from the signal itself.
- *Errors in the recorded performance:* To err is human, and even virtuoso musicians make many minor mistakes in every performance, and deviate to a greater or lesser extent from the original score. These “inaccuracies” help to give music its soul and serves to differentiate humans from machines². Eliminating these *human* errors is more difficult, as the system has to compensate for mistakes over which it has no control. Using probabilistic “soft” decisions based on a note’s meaning in the overall context of the performance instead of “hard” decisions based only on the note’s physical attributes might help to enhance the accuracy of transcriptions of human performances.

4.2 Restrictions

The scope of the complete automatic transcription problem outlined above is daunting. In order to reduce the scope of the problem for this exploratory thesis, a number of simplifications to the transcription problem were undertaken. The most important of these simplifications is the fact that the system architecture is purely bottom-up.

Furthermore, of the three major components for extracting physical data from the

²It is interesting to note that many modern sequencers, such as *Sibelius*, incorporate probabilistic mechanisms to slightly vary performance parameters such as rhythm so that the machine performance sounds more “natural”, “warm” and “artistic”.

waveform, only multi-pitch determination was implemented. The absence of a rhythm tracker has the following important disadvantages:

- *A fixed frame length:* The same frame length has to be used for fast and slow performances, and thus no dynamic trade-off between frequency and time resolution can be negotiated.
- *Overlap of notes that were played sequentially:* Each analysis frame is assumed to contain a stationary signal segment. Because the frames cannot be aligned with note onsets, some frames will blur the data across note boundaries because stationarity obviously is an invalid assumption at those points of the signal.
- *Loss of important information about the signal:* Rhythm is a crucial component of music. Without a separate rhythm analyser, much of this information cannot be extracted from the signal, making a sensible metrical grouping of notes impossible.

The absence of instrument models and classifiers also impacts on the effectiveness of the system:

- *Part separation is impossible:* It is impossible to group notes according to the instruments which generated them if the generating instruments are not determined. Thus the best that can be done is to recognise chord structures in the music.
- *Multi-pitch estimation is less accurate:* Without timbre models, notes with completely overlapping harmonics cannot be resolved, and notes with partially overlapping harmonics can only be resolved with reduced accuracy. This problem is discussed in more detail in the following chapter.

These limitations can be overcome to some extent by making assumptions about the signal or by attempting to extract the lacking data from the pitch tracker output. Such assumptions and techniques are discussed in the appropriate sections in the following chapters.

The higher-level processing capabilities of the system are also scaled down. The following parts of the system were designed and implemented to some degree of success:

- *Basic post-processing*
- *Key signature detection*
- *Estimation of the degree of polyphony*
- *Note duration quantisation*

However, a number of important processing components were not designed and experimentally investigated:

- *Analysis of the meter*

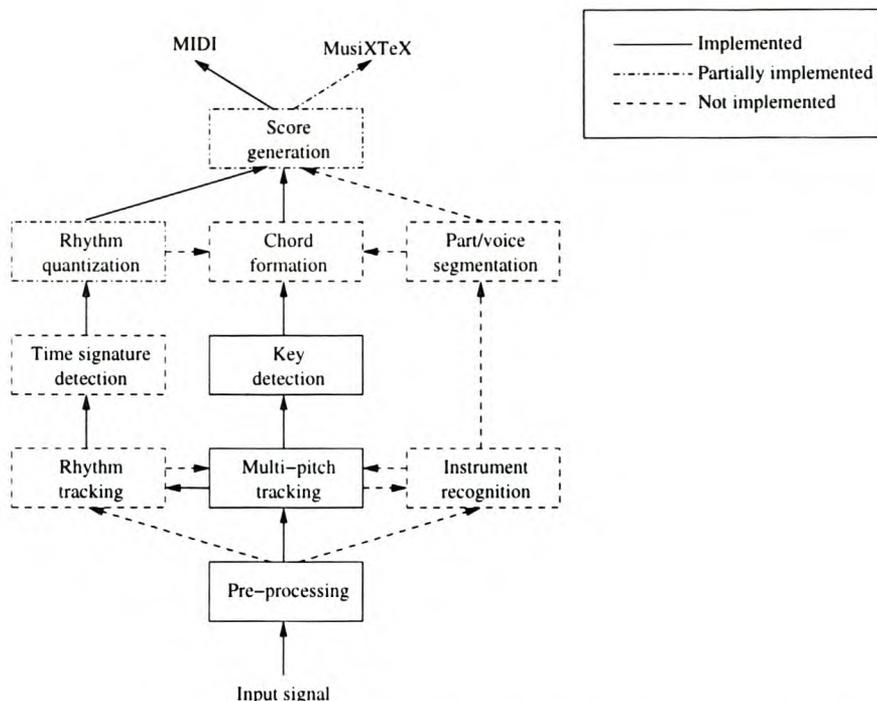


Figure 4.1: Breakdown of the AMADEUS transcription system

- *Time signature detection*
- *Assigning of notes to voices and parts*
- *Analysis of harmony*

As the final step of processing, the output of the system is written to MIDI files and MusiX_TE_X score files, with the restriction that the latter is only done for monophonic music signals and with the understanding that the accuracy of the output can obviously not rise above the limitations of the processing modules of the system. For example, the MusiX_TE_X scores cannot have coherent bar divisions if the system does not group the notes meaningfully into measures.

Figure 4.1 provides an overview of the transcription system, named AMADEUS (*Automated Music Analyser DEveloped at the University of Stellenbosch*) after the great composer W.A. Mozart³. The components with solid lines are implemented parts of the system, those with dash-dotted lines are only partially implemented for monophonic signals, and those with dashed lines are planned future extensions.

³The 12-year old Mozart is said to have transcribed Gregorio Allegri's "*Miserere Mei Deus*" from memory after one hearing, to the consternation of the Vatican which jealously guarded against the transcription of this musical treasure [10].

Chapter 5

Multi-Pitch Estimation

5.1 Background

This chapter documents the algorithms in the transcription system which transform the pre-processed waveform into raw notes. The two main steps in this process are pitch estimation on a frame-by-frame basis, and the integration of these pitch values into raw notes by tracking pitch across frames.

5.1.1 Importance and limitations

Pitch extraction is one of the most important steps in the transcription process, and it seems that many researchers have focused on developing multi-pitch estimation methods as the “Holy Grail” of automatic transcription. Even though the success of subsequent steps of processing hinges upon the success of the polyphonic pitch extractor, the structure of the music (harmony, harmonic progression, rhythm, etc.) itself provides a vast wealth of clues that can be used to eliminate unlikely combinations of pitch candidates.

The foregoing comment should in no way detract from the importance of this step of processing. It should just be kept in mind that satisfactorily solving the transcription problem in future will be determined ultimately by the development of successful musical models that employ various form of musical foreknowledge.

5.1.2 Requirements

The requirements of a multi-pitch estimator would be, at the very least, to (1) detect all component pitches (2) accurate to the nearest semitone (3) in the correct octave.

Fulfilling the first of these requirements is complicated by the nature of the polyphonic pitch determination problem. The ease with which the second criterium can be fulfilled depends on the mid-level representation from which the pitch is estimated; in the case

of the FFT, the inevitable frequency-time resolution trade-off makes it somewhat more difficult to determine the pitch of low notes accurately. The last requirement is to safeguard against harmonic errors, typically to an octave or twelfth above or below, due to incorrectly resolving a harmonic series to its fundamental. These problems, and ways to overcome them, will form the basis of much of the remainder of this chapter.

5.1.3 The Basic Pitch Determination Problem

Monophonic Pitch Determination

Fairly successful monophonic transcription systems have been in existence for several decades already. Nevertheless, it is a simplified instance of the more general polyphonic case and it is helpful to build the theory up from the monophonic case.

Assuming a single voiced instrument with partials that are harmonic, the time-discretised sound S can be described with:

$$S[n] = \sum_{h=1}^H A_h \sin(2\pi h f_0 n + \theta_h) \quad (5.1)$$

where $S[n]$ is the n -th sample in the steady-state portion of the signal, f_0 is the fundamental frequency (“pitch”) of the note, H is the total number of harmonics, and A_h and θ_h are the amplitude and phase of the h -th harmonic respectively. A_h and θ_h depend on the instrument, the note that is played and the loudness with which it is played, although they can be modelled to some accuracy for each note of an instrument. The specific series $\mathcal{A} = \{A_1, A_2, \dots\}$ contributes significantly towards the timbre of a particular instrument.

From the above, the spectrum of the note will have significant components at the frequencies:

$$f_h = h f_0 \quad 1 \leq h \leq H \quad (5.2)$$

with strengths at each f_h that are proportional to A_h .

Although a number of different monophonic pitch tracking strategies exist, for the purposes of this development the pitch tracking problem can be viewed as finding the f_0 for a specific time frame which best accounts for the significant spectral components.

If \mathcal{A} is compared to models of different instruments, the closest match can be taken to be the instrument I which produced the sound. This procedure then gives us values for pitch f_0 and instrument I at a specific point in the signal.

Polyphonic Pitch Determination

The pitch determination problem outlined above becomes a lot more involved when the sound S is a mixture of N_I different voiced instruments. The sound waveform at discrete time instant n can then be approximated by:

$$S[n] = \sum_{i=1}^{N_I} \sum_{h_i=1}^{H_i} A_{h,i} \sin(2\pi h f_{0,i} n + \theta_{h,i}) \quad (5.3)$$

Given any two random notes with fundamental frequencies f_p and f_q that are sounded together, the frequencies of any of their respective harmonics h and k will be given by:

$$f_h = h f_p \quad (5.4)$$

$$f_k = k f_q \quad (5.5)$$

The harmonic h of the note f_p will overlap with harmonic k of note f_q if:

$$f_h = f_k \quad (5.6)$$

Thus:

$$h f_p = k f_q \quad (5.7)$$

$$\Rightarrow f_p = \frac{k}{h} f_q \quad (5.8)$$

$$\Rightarrow \frac{f_p}{f_q} = \frac{k}{h} \quad (5.9)$$

From the above it can be seen that there will be overlapping harmonics in the spectrum of the composite sound if the fundamental frequencies f_p and f_q are ratios of integers of each other. In Section 3.3.2, it has been discussed that scales and chords are defined in such a way that intervals are given by ratios of small integers of the fundamental frequencies of notes. Thus when any two notes from standard scales are sounded together, there will be at least some overlapping harmonics.

As long as the two notes contain significant non-overlapping harmonics, the set of significant frequencies \mathcal{F} in a sound mixture can be resolved into the fundamental frequencies f_p and f_q that best account for the mixture.

However, if the fundamental f_q of one of the notes is located at any frequency

$$f_q = N f_p \quad \text{where } N \in \mathbb{N} \quad (5.10)$$

then any arbitrary harmonic h of f_q will be given by:

$$\begin{aligned} h f_q &= h N f_p \\ &= k f_p \end{aligned} \quad (5.11)$$

where $k = hN \in \mathbb{N}$ since it is the product of two natural numbers. The fact that every harmonic of f_q is an integer multiple of f_p implies that every harmonic of f_q coincides with a harmonic of f_p ! In other words, if f_q coincides with the frequency of a harmonic of f_p , then the spectra of the two notes overlap completely and the fundamental frequency f_q cannot be resolved by simply applying the criterion of harmonicity to the set of frequencies \mathcal{F} .

This then is the fundamental problem of multi-pitch estimation. In order to separate musical sounds with completely overlapping partials, different techniques and more knowledge about the signal have to be applied, including:

- Knowledge of the instrument partials
- Auditory cues as discussed in Section 3.1.2

However, these knowledge sources and cues do not form part of the current system. Their integration into the implementation is recommended as an important area of future research.

5.1.4 Mid-level representation

When designing a multi-pitch estimation algorithm, a mid-level representation needs to be chosen. The modern approach to multi-pitch estimation seems to lean towards correlation-based methods, as discussed in Section 2.3. However, for this thesis, a Fourier-based method was developed, for the following reasons:

- *The correspondence of the spectral representation with the physical signal:* Equation 5.3 established that voiced musical signals can be thought of as the sum of harmonically-related sinusoids. Significant peaks in a spectral-based representation can conveniently be determined and processed to provide the frequencies of the signal's component sinusoids. Other techniques such as orthogonal wavelets and correlation-based methods relate in more abstract and subtle ways to the physical signal. Given the exploratory nature of this work, it was deemed most appropriate to choose an analysis method which corresponds closely with the physical nature of musical signal.
- *Stability and predictable behaviour:* Unlike in the case of many spectral and pseudo-spectral estimation methods like ARMA and MUSIC (both of which were investigated during the initial stages of development), no assumptions concerning the number of component sinusoids need to be made for FFT analysis. The FFT thus gives very reliable estimates¹ of amplitude and phase of significant components across the

¹Within the time-frequency resolution framework of a chosen window type and analysis frame size

spectrum, making it fairly easy to extract data for all significant components.

The FFT has a fundamental flaw when processing musical signals though. As was shown in Section 3.3.2, musical scales are based on a geometric progression of frequencies. However, the FFT has a fixed frequency resolution Δf due to the linear frequency spacing of the bins. The frequency f_k of the k -th bin is given by:

$$f_k = k\Delta f \quad (5.12)$$

where the frequency resolution Δf of the FFT and the true frequency resolution Δf_{true} (the closest sinusoids that can still be separated) are given by:

$$\Delta f = \frac{f_s}{N_{FFT}} \quad (5.13)$$

$$\Delta f_{true} = \frac{f_s}{N_{win}} \quad (5.14)$$

This last equation points out another inconvenience of the FFT for musical processing. There is a Heisenberg-type trade-off between time resolution and frequency resolution because the product $\Delta f_{true} \times N_{win} = f_s$ is constant. The frequency resolution of the FFT can be improved through a combination of good peak-picking with the LULU algorithm described in detail in Appendix B and the frequency sharpening method described in Appendix C. Furthermore, if a sound has H prominent harmonic partials, the fundamental f_0 can be resolved to a resolution of $\frac{\Delta f}{H}$ because:

$$f_H - \Delta f \leq f_H \leq f_H + \Delta f \quad (5.15)$$

$$\Rightarrow Hf_0 - \Delta f \leq Hf_0 \leq Hf_0 + \Delta f \quad (5.16)$$

$$\Rightarrow f_0 - \frac{\Delta f}{H} \leq f_0 \leq f_0 + \frac{\Delta f}{H} \quad (5.17)$$

Thus the frequency resolution problem can be circumvented.

5.1.5 Synthetic signal

The algorithms given in this chapter were tested during development with synthetic signals, to ensure that the system works as expected. In order to illustrate the way the algorithms operate on data, results from one such signal will be used throughout the following sections. The demonstration signal is the first one-and-a-half measures of Michael Nyman's "*The Heart Asks Pleasure First*" from the soundtrack to the movie *The Piano*:



The sample was synthesised in MATLAB by summing sinusoids of appropriate durations at various pitch and harmonic frequencies, according to Equation 5.3. The component synthetic sounds all have the same harmonic structure \mathcal{H} , as shown in Figure 5.1, and the sounds were all synthesised with the same loudness. These simplifications were necessary to overcome some of the inherent design limitations, as discussed in Section 5.4. The fact that the notes were strictly mixtures of sinusoids and with all notes having identical loudnesses made the polyphonic signal sound like a fairly realistic organ sample. The sample was synthesised with 50 ms breaks between repeated notes to separate them. The tempo was chosen as $\downarrow = 50$, so that quavers have a duration of 0.4 s. It should be noted that this sample has a degree of polyphony of four.

A spectrogram of the synthetic signal is given in Figure 5.2. Note that the signal contains up to three simultaneous notes with completely overlapping component spectra (the fourth from last chord has three simultaneous A's in different octaves). This property of the sample enables for a very real demonstration of the multi-pitch estimation problem defined in Section 5.1.3.

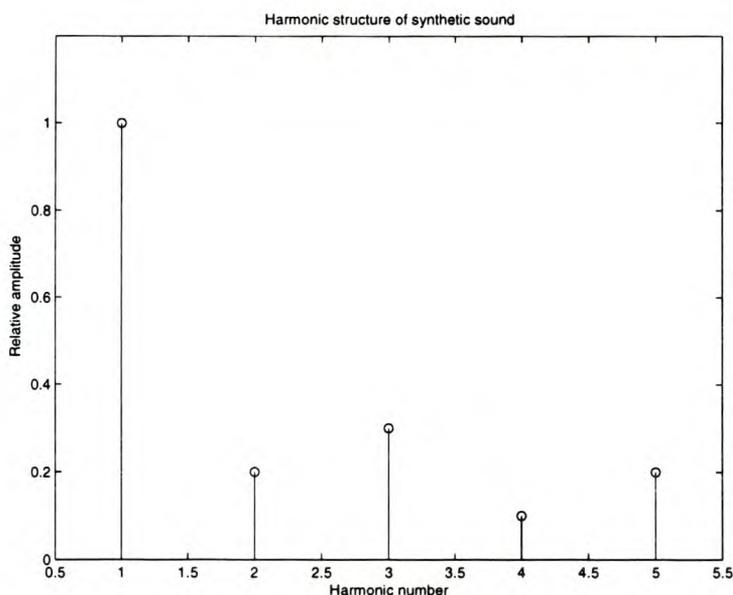


Figure 5.1: *Harmonic structure of synthetic “instrument” sound*

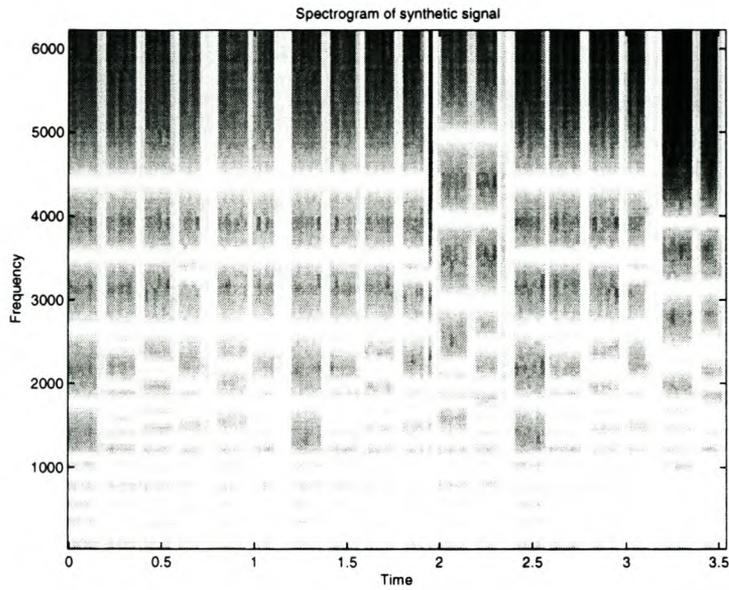


Figure 5.2: *Spectrogram of synthetic polyphonic sample*

5.2 Pitch estimation algorithm

In this section an algorithm for an FFT-based multi-pitch estimator is developed and discussed. The transcription problem was approached by first developing a robust monophonic pitch tracking method with which the simpler case of monophonic transcription was investigated. The current multi-pitch estimation algorithm is an expansion of the original monophonic pitch estimator².

5.2.1 Algorithm outline

The pitch detection algorithm can be outlined as follows:

²Care was taken to ensure that the algorithm reduces to monophonic pitch estimation for monophonic signals.

- Step 1: FFT** The FFT is calculated for each time frame, with windows 46.4 ms long and with 87.5% overlap between frames. Square windows (boxcar windows) are used to achieve the narrowest possible main lobes.
- Step 2: LULU** The power spectrum is then filtered with a non-linear impulse transformation (refer to Appendix B) of order $N_{LULU} = 1$ to remove the FFT sidelobes. Peak-picking is subsequently performed on the spectrum to find significant frequencies \mathcal{F} .
- Step 3: Frequency sharpening** The frequencies of found peaks are then sharpened with a technique borrowed from phase vocoding (refer to Appendix C), assuming a sinusoid at each peak.
- Step 4: Determine Candidates** A list of pitch candidates is calculated from the significant frequencies obtained in Steps 2 and 3.
- Step 5: Score candidates** Candidates are scored by summing the power contained in each partial.
- Step 6: Validate candidates** Candidates are validated using a number of heuristics. The validated candidate with the highest score is added to the list of pitch mixture components for the current frame.
- Step 7: Remove partials of highest scoring candidate from list** The partials of the highest scoring valid candidate are removed from the list \mathcal{F} of significant frequencies.
- Iterate Steps 4-7** The procedure is iterated until no significant contributing pitch candidates can be found in the current frame.

These steps will now be illustrated and discussed in greater depth in Sections 5.2.2 to 5.2.6.

5.2.2 Determining spectral peaks

When the signal is FFT'ed, the window size is chosen so as to give a true frequency resolution of around 20 Hz at a sampling rate of 44.1 kHz. This provides a window size of 2048 samples or 46.4 ms. Because frequency resolution is very important for resolving low notes, square windows are used to achieve the narrowest possible main lobes. The frequency response of the square window has main lobes that are $R = \frac{N_{FFT}}{N_{win}}$ bins wide, with side lobes that are half as wide. Unfortunately, this good frequency resolution is accompanied by a peak side lobe which is only 13 dB below the main lobe [47, p. 628]. The

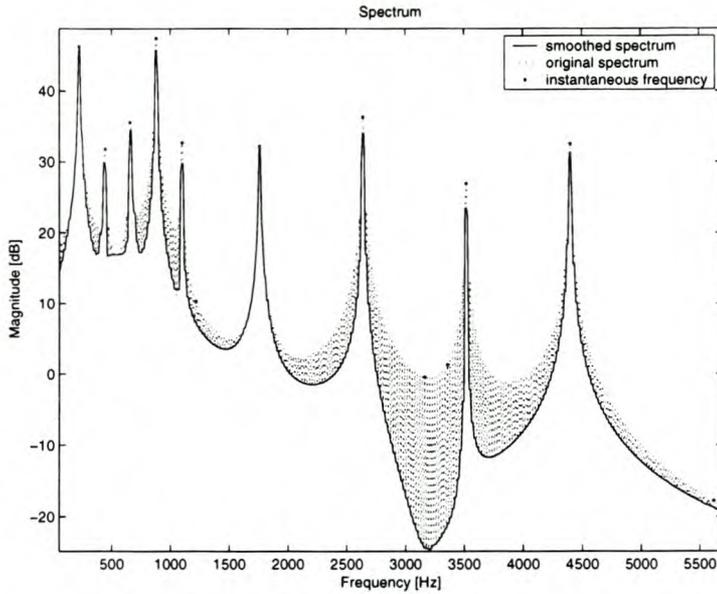


Figure 5.3: *Peak picking on the spectrum of a synthetic polyphonic sound*

side lobes are suppressed by applying a non-linear impulsive smoother to the spectrum. Such a *LULU* smoother of order N_{LULU} suppresses impulsive artifacts narrower than $N_{LULU} + 1$. Thus the smoother’s degree is chosen as $N_{LULU} = R - 1$ in order to “filter out” the sidelobes.

Assuming that spectral peaks were generated by sinusoids, the true frequency of the generating sinusoid can be determined by examining the phase difference over two frames of the FFT bin which contains the peak. The result of this series of operations (FFT, *LULU*-smoothing, peak-picking and frequency sharpening) is a list of significant frequencies \mathcal{F} for the current frame.

An example of this process for one frame of the synthetic demonstration signal is given in Figure 5.3. It can be seen that all peaks are correctly identified (even though there are a few spurious ones also). The effect of the *LULU*-smoother on the spectrum is quite remarkable, suggesting that such smoothers are very effective for removing outliers from data. It can also be seen that the peaks are sharpened correctly: the instantaneous frequencies are given more accurately than the actual spectral resolution.

5.2.3 Determining pitch candidates

The relationship between the fundamental frequency and its harmonics is used to determine possible pitch candidates from the list of spectral peaks. Each peak $f_{peak} \in \mathcal{F}$ gives

a number of pitch candidates by iterating through $h = \{1, \dots, 5\}$ to calculate:

$$f_{0,cand} = \frac{f_{peak}}{h} \quad (5.18)$$

provided that $f_{0,cand}$ lies in the range of 80 to 2500 Hz (E_2 to E_7).

This procedure generates a lot of duplicate pitch candidates because many harmonics will resolve to similar fundamentals. However, the above procedure is used to ensure that pitch candidates are generated even when a number of peaks have been missed by the peak picking procedure. It also ensures better frequency resolution when frequency sharpening was unsuccessful.

The duplication of pitch candidates (as well as the generation of invalid pitch candidates) in the above procedure is not problematic because all candidates are examined and validated in subsequent steps of processing.

5.2.4 Candidate scoring

The pitch candidates are “scored” by calculating the total candidate power $\mathcal{P}_{tot,cand}$ by summing the powers $\mathcal{P}(f)$ at the harmonics frequencies f of each pitch candidate:

$$\mathcal{P}_{tot,cand} = \sum_{h=1}^H \mathcal{P}(hf_{0,cand}) \quad \text{where } H = \left\lfloor \frac{f_s/2}{f_0} \right\rfloor \quad (5.19)$$

The normalised harmonic structure $\mathcal{K} = \{K_1, K_2, K_3, \dots, K_H\}$ of each candidate is also determined:

$$K_h = \frac{\mathcal{P}(hf_{0,cand})}{\max \mathcal{P}(kf_{0,cand})} \quad \text{where } 1 \leq h \leq H \quad (5.20)$$

This harmonic structure is used to validate candidates in the following step. In future versions of the system \mathcal{K} can also possibly be used for instrument identification by matching it to known (trained) values for various instruments.

5.2.5 Candidate validation

The pitches and their power scores are then used to validate the candidates. Candidates must satisfy the following conditions:

1. They must either have a strong fundamental ($K_1 > 0.05$),

or

They must have strong second to fifth harmonics ($K_h > 0.05$, where $2 \leq h \leq 5$). These criteria are meant to ensure that the combination of harmonics of the candidate stimulates a strong sensation of pitch at the candidate frequency.

2. The odd partials must contribute significantly to the total power of the note:

$$\sum K_{odd} \geq 0.2 \sum K_{even} \quad (5.21)$$

This is to reduce the occurrence of downward octave errors, where the candidate fundamental is half the true fundamental, implying that the odd harmonics of the incorrect candidate will all have relatively low amplitudes.

3. The total candidate power $\mathcal{P}_{tot,cand}$ must constitute a significant fraction of the total power \mathcal{P}_{tot} in all the found peaks: $\mathcal{P}_{tot,cand} \geq 0.2\mathcal{P}_{tot}$.

All of the above validation criteria are heuristically founded on the spectra of a variety of instrumental and vocal sounds. Nonetheless, a better strategy would be to compare the harmonic structure \mathcal{K} of each candidate with spectral models of instruments at that pitch to determine whether the candidate \mathcal{K} is valid in terms of the models. However, the required instrument models do not currently form part of the system. The listed validation criteria can also not be guaranteed to produce results that correspond exactly with the *physical mixture* of notes that produced the compound sound in a given frame. Instead, these criteria were chosen to provide results that are a reflection of the *perceptual mixture*, which is the best that can be done without a knowledge source for instrument timbres.

Sounds which were generated on instruments like the violin and the oboe typically have partial structures where the fundamental is relatively weak compared with some of the upper partials. In such cases the separation of notes with completely merged spectra is impossible without using instrumental models and a variety of auditory streaming cues.

In some special cases, however, notes with merged spectra can be separated, if the following assumptions are made:

1. The fundamental is the strongest partial of a note's response.
2. Notes that are sounded simultaneously have similar intensities, or less strictly: notes of higher pitch have greater intensities.

Many notes on the piano, acoustic guitar and other plucked string instruments, along with the organ, flute, recorder and other air reed instruments have dominant fundamentals, especially for notes above Middle C. The synthetic signal also fulfills both of the above requirements. In such cases, merged notes can be separated with the criterium:

$$K_h > K_1 \quad \Rightarrow \quad hf_0 \text{ is a separate note.} \quad (5.22)$$

5.2.6 Removing partials from the partial candidate list

The highest scoring *validated* pitch candidate $f_{0,max}$ is added to the pitch list for the current frame. In order to find further notes in the current frame, all partials belonging

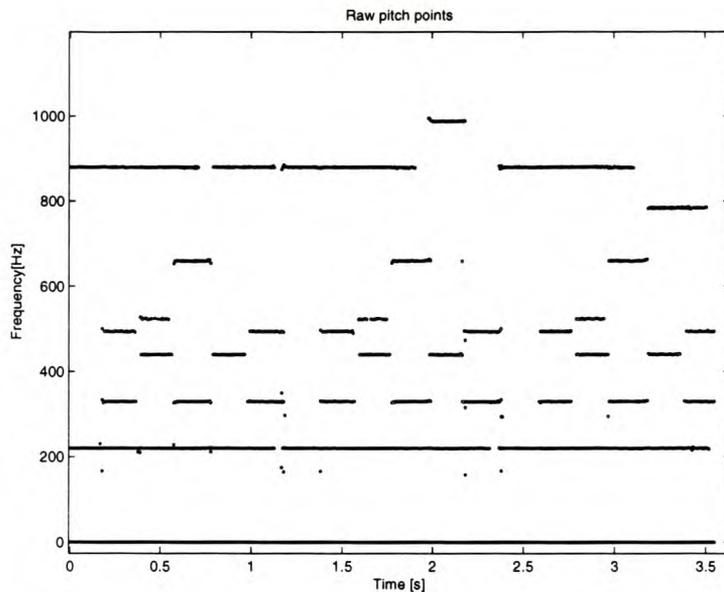


Figure 5.4: *Raw pitch points of synthetic polyphonic sample*

to $f_{0,max}$ are removed from the list of significant frequencies \mathcal{F} . In order to account for some degree of inharmonicity in the partials, a frequency $f_{peak} \in \mathcal{F}$ is removed if an h can be found such that:

$$0.98hf_{0,max} \leq f_{peak} \leq 1.02hf_{0,max} \quad (5.23)$$

The factor 1.02 is chosen because it is somewhat less than half a semitone ($2^{1/24} = 1.0293$) and partial candidates falling in that range can thus be assumed to belong to $f_{0,max}$.

With the reduced list of partial candidates, the process of finding, scoring and validating pitch candidates is iterated again, until such time as no more viable pitch candidates can be found.

The result of the pitch estimation algorithm for the synthetic signal is given in Figure 5.4. It can be seen that virtually all pitch points (including those with completely merged spectra that had to be detected with the condition in Equation 5.22) were found correctly, with only a number of small errors at the note onsets and endings. The pitch of 0 Hz that is given in every frame serves as a terminating value for each frame's pitch list.

5.3 Pitch tracking algorithm

The above pitch estimation procedure results in a list of pitch points for each frame. The pitch points from successive frames are then grouped into notes across time frames. The

pitch tracker provides a set of “raw” notes with the properties:

- N_{start} : the analysis frame in which the pitch was first encountered
- L_{note} : the length of the note in number of analysis frames
- M : pitch specified as a MIDI semitone
- \mathcal{P}_{avg} : the average power of the sound

5.3.1 Algorithm

The pitch tracking algorithm is outlined as follows:

Step 1: Convert all pitch points from Hz to MIDI semitones using Equation 3.15.

Step 2: For all frames $1 \leq N_{cur} \leq N_{frames}$, do step 2.1:

Step 2.1: For all pitch values in frame N_{cur} that have not been marked “processed”, do steps 2.1.1-2.1.2:

Step 2.1.1: Find pitch points at the same pitch in the following frames $N_{next} > N_{cur}$, not skipping more than N_{skip} frames between each two frames with matching pitch points. All pitch points in successive frames that are thus found need to be marked as “processed”. Let N_{last} be the last frame that contains a matching pitch point and $N_{start} = N_{cur}$. Then the note length $L_{note} = N_{last} - N_{cur} + 1$.

If at any time during the described forward searching there is an increase in the sound power of more than a factor 10 from a matching pitch point to the next, stop the forward search at that frame (which becomes N_{last}). Such a power increase signals the attack of a new note.

The average note power \mathcal{P}_{avg} is the mean of the individual powers of the matching pitch points.

Step 2.1.2: If $L_{note} > L_{min}$, add it to the list.

A sensible minimum note length L_{min} in the above algorithm would be the equivalent of a duration of around 60 ms, which is the duration of a demisemiquaver () played at $\text{♩} = 120$. N_{skip} can be assigned half the value of L_{min} .

Pitch tracks obtained with this algorithm are given in Figure 5.5 for the synthetic signal. It can be seen that all notes given in the original score are present in the pitch tracks, and no incorrect notes were detected.

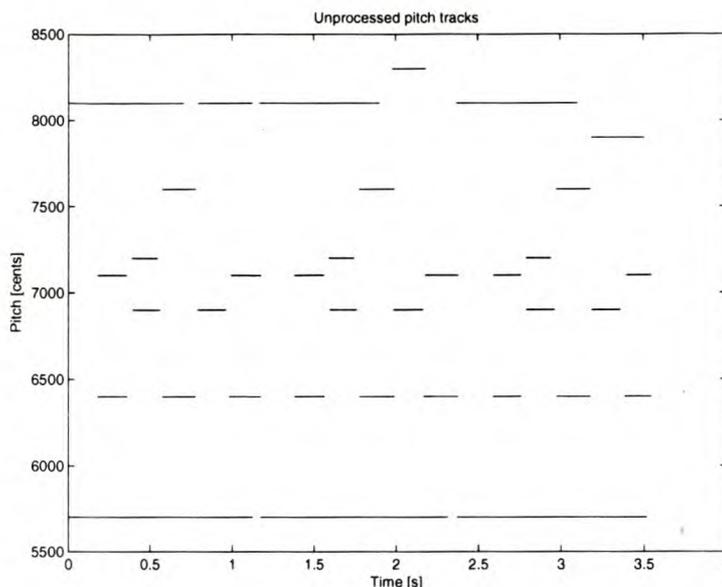


Figure 5.5: *Unprocessed pitch tracks of synthetic polyphonic sample*

5.3.2 Discussion

The described pitch tracking algorithm makes two restrictive assumptions about the signal:

- *Instruments are tuned close to the standard:* The instruments which generated the sounds are tuned in such a way that no note is more than 50 cents off A440-based equal-tempered pitch. If a note is more than 50 cents off standard equal-tempered pitch, it will be discretised to the wrong semitone in Step 1.
- *Vibrato depth is restricted:* Although the fact that the system tracks notes with semitone resolution allows for a relatively deep vibrato, a frequency deviation of greater than a semitone (peak-to-peak) will show up in the note tracks as an oscillation between several adjacent semitones.

The algorithm proposed above is not well-suited for transcribing vocal music. Unaccompanied singers will generally have a (more or less constant) offset to standard pitch, due to the fact that most humans do not have perfect pitch. This offset needs to be determined and subtracted from the pitch (in logarithmic cents). Moreover, many singers have a very deep vibrato (we have measured depths of a whole tone peak-to-peak for some singers!) which obviously violates the vibrato depth assumption of the algorithm.

Both of these restrictions have been overcome in one of our prototype monophonic transcription systems by segmenting the signal with an algorithm that has a certain “inertia” which favours regions where the *mean* pitch is stable for a certain period of time. The mean offset of the stable regions to their nearest semitone can then be subtracted

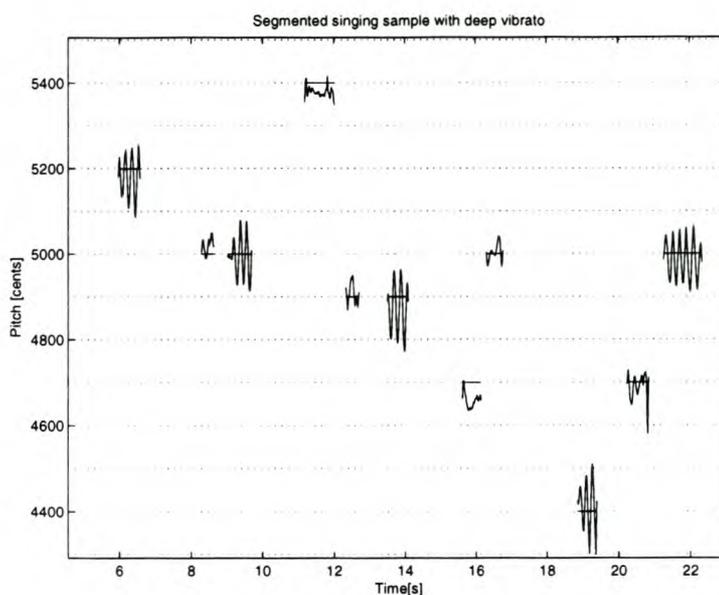


Figure 5.6: *Singing sample with deep vibrato*

from the pitch curves to shift them to standard chromatic pitch. The stable regions are assumed to be notes with a pitch equivalent to the mean of the region. This algorithm was found to work well for monophonic singing samples. An example of a sample with deep vibrato as segmented by this alternative algorithm is given in Figure 5.6. The mean offset to standard tuning has already been removed from the given pitch track.

5.4 Limitations

The multi-pitch estimation and tracking method proposed in this chapter has a number of inherent limitations. The absence of tone models in the system is probably the most severe limiting factor. Because of this, it is impossible to separate fused tones. Even more fundamentally, without knowledge of what a specific instrument’s spectrum for a certain pitch looks like, it is impossible to say with certainty whether a pitch candidate together with the magnitudes of its harmonics actually constitutes a valid component sound for a given mixture.

The multi-pitch estimation algorithm makes explicit use of only one of Bregman’s auditory cues, namely that of *harmonicity*. It is imperative that future systems make use of at least the *common onset* cue in addition to harmonicity. Using common onset, notes with overlapping spectra but different onset times can be separated by always grouping only new frequencies that were absent in the previous frame. We attempted to integrate a version of the “stream tracers and watchers” algorithm in Figure 3.3 into the pitch estimation algorithm. This extension component added all validated pitches from the

previous frame to the list of pitch candidates in the current frame. However, it was found that this sometimes led to an unstable pitch tracker which generated an ever-increasing number of “ghost pitches”.

Chapter 6

Post-Processing

6.1 Introduction

As important, or perhaps even more important, than an accurate pitch tracker is a post-processing module that transforms the raw notes into usable results through the application of different rules.

6.2 Basic post-processing

The results of the pitch tracker are sometimes very rough in that many spurious notes may have been found. In this section, algorithms are presented which are used in the current transcription system to weed out two types of such notes: those with a low average power (soft notes) and those that exceed the estimated degree of polyphony of a piece.

6.2.1 Eliminating notes with low power

Soft background noises (coughs, air conditioners) during silences in the music are often detected by the pitch tracker as notes. Note that such noises are not generally detected in segments with true sound content because of the restriction that every note in a certain frame must constitute some significant fraction of the total power. To eliminate the “noise notes”, the power values of the notes are sorted into an array, which is then partitioned with the Lloyd-Max quantisation algorithm (see Appendix D) into two groups: *loud notes* and *soft notes*. If there is a sufficient distinction between the *loud* partition and the *soft* partition, the notes in the soft category are eliminated from the note list.

The detail of the algorithm is as follows:

- Step 1:** Cluster all values of average note power in dB ($10 \log_{10} \mathcal{P}_{avg}$) into two partitions having cluster means μ_1 and μ_2 respectively. The logarithm of power is used so as to allow for the fact that sound levels during a performance may vary considerably.
- Step 2:** Ensure that $|\mu_2 - \mu_1| > 20$ dB. If not, terminate the algorithm. This ensures that valid notes are not eliminated in a signal that is mostly free of background noise.
- Step 3:** The threshold for minimum valid note power is set at $\mathcal{P}_{min,dB} = \frac{1}{2}(\mu_1 + \mu_2)$, the boundary between the clusters. Eliminate all notes with power lower than this threshold.

In effect, this method chooses a dynamic (as opposed to a hard-coded) threshold as its elimination criterium. This is necessary because the approximate loudness of the component sounds can vary considerably from sample to sample. Furthermore, using two centroids instead of the single mean is necessary if an appreciation of the spread of values is to be gained.

6.2.2 Detecting polyphony

It is useful to detect the degree of polyphony of a piece. For example, this allows for elimination of excess simultaneous notes.

An approximation of the degree of polyphony is easily obtained using the following strategy:

Step 1: Obtain a list of note events (note starts and note endings), sorted according to the instants at which they occur.
 Set some variable **curpoly** \leftarrow 0.
 Initialise an array **polyhistogram**(•) of indefinite size to zero.

Step 2: For all note events, do steps 2.1-2.2:

Step 2.1: If the current event is a note start, set **curpoly** \leftarrow **curpoly** + 1.
 If the current event is a note stop, set **curpoly** \leftarrow **curpoly** - 1.
 Also set $\Delta T \leftarrow$ (instant of current event) - (instant of previous event)

Step 2.2: Set **polyhistogram**(**curpoly**) \leftarrow **polyhistogram**(**curpoly**) + ΔT

Step 3: Calculate the cumulative sum **polycumsum** \leftarrow *cumsum*(**polyhistogram**)

Step 4: The first bin **degpoly** for which
polycumsum(**degpoly**) \geq 0.8 max **polycumsum**
 gives the approximate degree of polyphony.

In the current transcription system, the degree of polyphony is used to eliminate the softest notes in segments where the degree of polyphony exceeds **degpoly**. Another more sophisticated use could be to use the approximate degree of polyphony to detect and correct the absence of notes in chords. For example, if a certain chord consists of only two notes, but the degree of polyphony was found to be three, then it is possible that the multi-pitch estimator missed one of the component notes.

For the synthetic demonstration signal, the program generated following output whilst calculating the degree of polyphony:

```
---POLYPHONY ELIMINATION---
0 simultaneous voices: 0%
1 simultaneous voices: 0.490998%
2 simultaneous voices: 24.2226%
3 simultaneous voices: 28.6416%
4 simultaneous voices: 46.4812%
Eliminate notes when >4 voices...
ORIG # NOTES = 36, NEW # NOTES = 36, # NOTES ELIMINATED = 0
```

The degree of polyphony was thus detected accurately. Because there were no incorrect insertions, no notes were eliminated.

6.3 Key Determination

6.3.1 Introduction

Determining the key of a transcribed piece is of cardinal importance for further processing. Many forms of harmony employ chords and sequences of chords that have specific meaning according to the degree of the scale on which they are constructed. Although musical models are not used for the purposes of this thesis, a simple yet effective algorithm for determining the key of a piece is developed.

6.3.2 Algorithm

The approach chosen for the current system employs circular convolution to determine the key signature (as opposed to the key itself, which can be either major or minor). In Section 3.3.3 and Table 3.3, we have seen that there is a correlation between major and natural minor scales with regards to their semitone intervals, even though they differ in their choice of tonic. In general, music written in a minor mode contains both raised and natural $\hat{6}$ and $\hat{7}^1$. However, this does not impact on the choice of key signature as the intervals for the remaining scale degrees are the same for all types of major and minor scales; these intervals are used to find the key signature. Finding the key signature can thus be defined as the problem of finding the pitch class which acts as the base for an “averaged” scale pattern which is valid for both major and minor keys.

First, histogram tallies $V(C)$ are calculated for occurrences of notes in each of the twelve pitch classes $0 \leq C \leq 11$. Pitch class C is determined from semitone M as:

$$C = M \bmod 12 \quad (6.1)$$

so that $C = 0$ represents C, $C = 1$ represents C \sharp , etc.

The tallies $V(C)$ are then circularly convolved [47, p. 416] with a model $U(C)$ which is a combination of the major and minor scales, as shown as the bottom graph in Figure 6.1. The pitch class C_{base} is taken to be the maximum value of this circular convolution W . This process is best described by the following equations:

$$W(C) = U(C) \circledast V(C) \quad (6.2)$$

$$= \sum_{j=0}^{11} U(C)V((C+j) \bmod 12) \quad \text{where } 0 \leq C \leq 11 \quad (6.3)$$

$$C_{base} = \arg \max W(C) \quad (6.4)$$

¹The notation \hat{n} indicates the n -th scale degree.

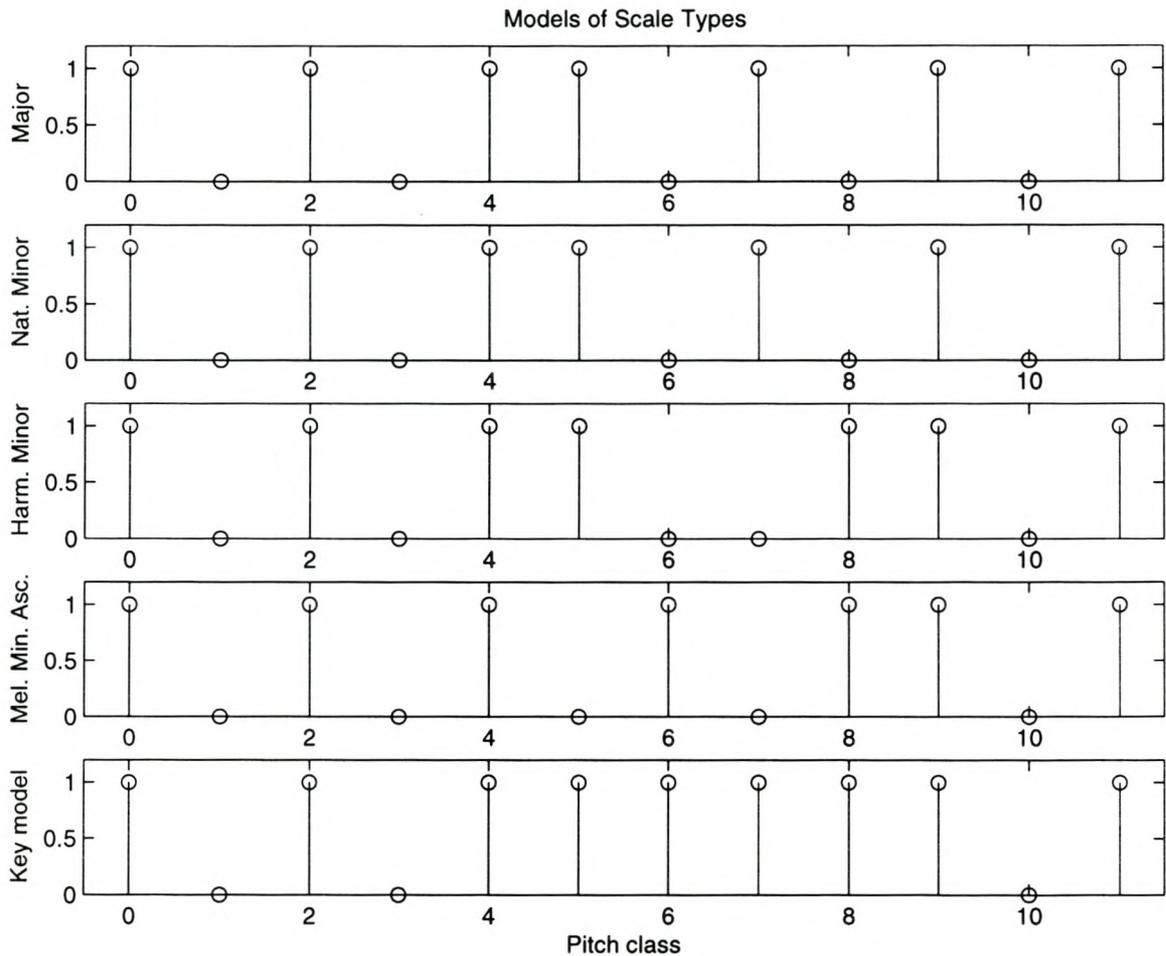


Figure 6.1: Models of various scale types: Shown from top to bottom are the C major scale, the corresponding natural a minor scale (which is also the pattern for the descending melodic minor scale), the harmonic a minor scale, the descending melodic a minor scale and the combination model which is used for key signature detection.

The key signature can be derived from C_{base} by using the circle of fifths in Figure 3.8. For example: $C_{base} = 0$ indicates a key of C major or corresponding a minor and thus the piece has an empty key signature; $C_{base} = 1$ indicates a key of $C\sharp$ or Db major (or their corresponding minors) and thus the signature is given by 5 flats (with the assumption that an “emptier” key signature is easier to read than one with more accidentals); etc.

A further condition is needed to distinguish between major and minor scales. Given a key signature, the way to determine the scale type (major or minor) is to determine the significant presence of raised minor $\hat{7}$ which is characteristic of music in minor modes (minor $\hat{7}$ corresponds with $\hat{5}$ of a major scale, a note which is generally not consistently raised). The pitch classes $C_{5,unraised}$ and $C_{5,raised}$ for natural and raised major $\hat{5}$ respectively are given by the following equation for an earlier determined C_{base} :

$$C_{5,unraised} = (C_{base} + 7) \bmod 12 \quad , \quad C_{5,raised} = (C_{base} + 8) \bmod 12 \quad (6.5)$$

Then, if

$$V(C_{5,raised}) > 0.4V(C_{5,unraised}) \quad (6.6)$$

the scale type is taken to be minor, else it is taken to be major.

6.3.3 Complete algorithm

The complete algorithm is summarised as follows:

- Step 1:** Calculate tallies $V(C)$ for the pitch classes C of all notes.
- Step 2:** Match $V(C)$ to the “averaged” scale model $U(C)$ with circular convolution: $W(C) = U(C) \circledast V(C)$. The best match is: $C_{base} = \arg \max W(C)$.
- Step 3:** Look up the key signature for a major scale based on C_{base} using the circle of fifths (Figure 3.8).
- Step 4:** Detect the scale type based on the possible presence of a large proportion of raised notes five scale degrees above C_{base} .

Although the multi-pitch estimator is not very accurate, the key was detected correctly for all real samples that were tested. This suggests that the multi-pitch estimator is accurate enough to allow for key detection. This begs for the implementation of top-down processing structures to use the key for enhancing the accuracy of multi-pitch estimation.

6.3.4 Discussion

For the synthetic input signal, the following output was generated:

```
---DETERMINING KEY---
noteTally[0] = 3
noteTally[1] = 0
noteTally[2] = 0
noteTally[3] = 0
noteTally[4] = 12
noteTally[5] = 0
noteTally[6] = 0
noteTally[7] = 1
noteTally[8] = 0
noteTally[9] = 13
noteTally[10] = 0
noteTally[11] = 7
=> C major
```

The internal procedure which is followed to determine the key is visualised in Figure 6.2. It can be seen that the key signature is ambiguous if the circular convolution method is used, because only 5 pitch classes are involved in the first few notes of the piece. The program chose the first pitch class with the highest value in the convolution result as the key signature, which gives *C* major / *a* minor. The distinction between major and minor is trivial here because there are no raised minor $\hat{7}$ in the extract, suggesting a major scale. However, even though there are no raised minor $\hat{7}$, the context of the music suggests a minor scale. Thus, for this 3.6 second short synthetic sample, the key signature was identified correctly only by chance from a set of 9 viable candidates. Furthermore, the scale type was misidentified. The lesson learned from this is that the key detection algorithm needs a larger and more representative data set than the 36 notes with which it was presented here. Additionally, it should be kept in mind that Nyman's music falls somewhat outside of the classical music theory on which the key detection algorithm was based.

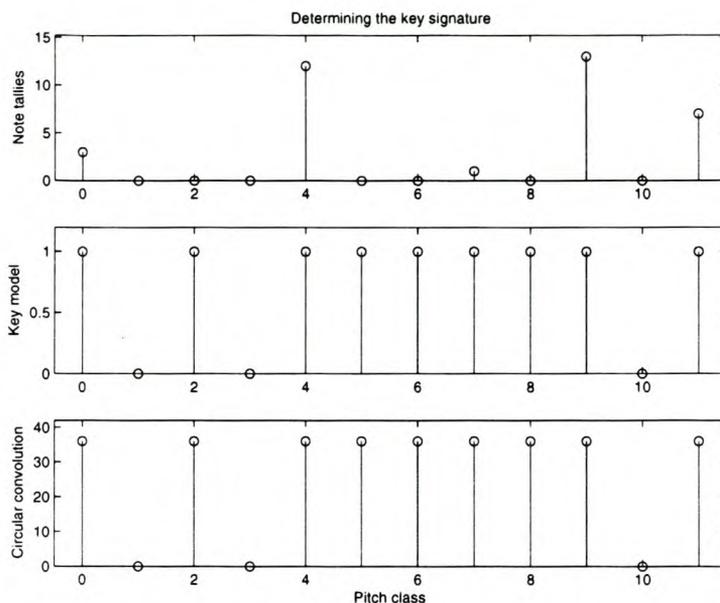


Figure 6.2: Visualisation of key signature detection for the synthetic signal

6.4 Note duration quantisation

6.5 Overview

As discussed earlier, the current system does not include a separate beat and meter tracking component. For score generation, however, the note durations need to be quantised in some or other fashion. For that reason a *least squared error* quantisation algorithm was devised.

This algorithm involves finding a base note duration (a “beat”), to which note durations are related through some factor. It should be noted that the “beat” as it is found and used here is not necessarily the true beat of the piece.

In Section 3.3.4, it was mentioned that the basic musical note durations differ from each other by factors of 2, although virtually any duration can be simulated with the use of grouplets, dotted notes and slurred notes.

The beat duration T_{beat} is related to the tempo B (typically expressed as beats per minute) as follows:

$$B = \frac{60}{T_{beat}} \quad (6.7)$$

The beat should be chosen within a sensible range. Standard metronomes generate tempos in the range of 40-208 bpm. The symbol (\downarrow , \downarrow , \downarrow etc.) chosen for the beat is

somewhat arbitrary, and a piece notated with a beat = \downarrow could just as easily be notated with beat = \downarrow (and all note duration symbols downscaled appropriately by half).

The algorithm for note duration quantisation works as follows:

The logarithms of the durations of the notes found by the pitch tracker are clustered into six groups using Lloyd-Max quantisation. The logarithms are used because note durations are typically related to each other by powers of 2. Six groups are used because there are six simple note durations from \downarrow to \circ . This provides six cluster centroids μ_j , where μ_j denotes the mean log-duration of notes in cluster j .

The note duration T_j represented by each μ_j can be calculated by

$$T_j = 10^{C_j} \quad (6.8)$$

Each T_j is then scaled to a value in the range of $0.375 \text{ s} \leq T_j \leq 1.2 \text{ s}$ (the equivalent tempo range of 50-160 bpm).

The ratios between the note duration T_{note} and the beat duration T_{beat} (which is assumed to be a notational quarter note) are calculated for each note using

$$R_T = \frac{T_{note}}{T_{beat}} \quad (6.9)$$

The closest match \widetilde{R}_T in the set of “standard” note durations $\mathcal{R}_0 = \{\frac{1}{8}, \frac{1}{4}, \frac{1}{2}, 1, 2, 3, 4, 6, 8\}$ is determined for each R_T . The values in \mathcal{R}_0 represent the factors by which certain note durations are related to *crotchets* (quarter notes). Thus a value of 1 indicates a quarter note, whilst a value of 4 is used for whole notes, $\frac{1}{8}$ for 32nd notes, etc. The set \mathcal{R}_0 is chosen in such a way that simple note durations are favoured above triplets: grouplet values have been omitted from \mathcal{R}_0 . This is so that true compound beats will not end up being notated with triplets.

The error for note j is then calculated as

$$e_j^2 = \left(\frac{\widetilde{R}_T - R_T}{\widetilde{R}_T} \right)^2 \quad (6.10)$$

The mean squared error is then

$$\overline{e^2} = \frac{1}{N_{notes}} \sum_{j=1}^{N_{notes}} e_j^2 \quad (6.11)$$

Finally, note durations are quantised to standard musical duration values using the T_{beat} which gives the lowest discretisation error. The scaling factors of T_{beat} which are used for duration quantisation are:

$$\mathcal{R} = \left\{ \frac{1}{8}, \frac{1}{4}, \frac{1}{3}, \frac{1}{2}, \frac{3}{4}, 1, \frac{3}{2}, 2, 3, 4, 6, 8 \right\} \quad (6.12)$$

The discretisation is done according to the procedure described above for finding the closest \widetilde{R}_T .

6.5.1 Algorithm

The note duration quantisation algorithm can be summarised as follows:

- Step 1:** Cluster the note durations into six clusters.
- Step 2:** Scale the cluster centroids to fall within a range of sensible “beat” durations.
- Step 3:** For each centroid, find the mean square error of note duration quantisation.
- Step 4:** Quantise to the “beat” duration which gives the smallest MSE.

6.5.2 Example

For the synthetic signal, the following output was produced:

```
Best fit base note duration (quarternote) = 0.380227 secs  
Tempo = 157 bpm
```

The system did indeed find the most sensible base note duration. The piece was generated at 50 bpm with dotted crotchets, which give quaver durations of 0.4 s. However, some notes were somewhat shortened (by 50 ms) to prevent repeated notes from merging. Thus the base note duration is a little bit shorter than the true quaver length.

6.5.3 Limitations

The above method, even if it were accurate in finding the true beat of a piece, is insufficient to provide information about the phase of the beat².

Currently, note durations are only quantised for monophonic music so as to allow for score generation with standard duration symbols. From the results in Chapter 8 it can be seen that even such a crude quantisation provides results that are intelligible enough to gain an impression of the original rhythmic phrasing.

Future versions of the system should implement ways to quantise note durations to values that are musically meaningful in the context of a phrase. For example, discretisation to triplet values should only be done if there are two or three notes that can be grouped into a valid triplet structure. This requires accurate estimates of the tatum, the

²By definition, the beat has a certain periodicity, and thus has a *frequency* and a *phase*. The phase describes at which fraction of the beat a certain note is located. Thus integer values of the phase correspond with the locations of the beat pulse [15].

beat and the measure of the signal. These estimates could then be used together with probabilistic meter grouping models for results that conform more closely to standard rhythmic notation.

6.6 Output

The last step performed by the transcription system is to generate output for all the notes which have been detected and processed. Most transcription systems generate MIDI output – a standard which forms the basis of computer music processing and data exchange, as discussed in Appendix E. MIDI files can be played with freely available players on all computer platforms, and are thus a convenient way to provide output that can be compared *aurally* to the original. Moreover, MIDI files can be imported into many score editing software packages. In fact, some researchers have chosen to write their transcription output exclusively to MIDI files, and leave the nitty-gritty of score generation (such as choosing suitable note groupings, beams, slurs, clefs, etc.) to commercial score editing packages. The problem with this is that MIDI files contain information about *physical* notes (pitch, start, end, velocity) together with some optional information (key signature, time signature, tempo) but do not store information about how individual notes should be notated.

For this reason, a first attempt at score generation for monophonic music was undertaken, in addition to MIDI output. MusiX_{TEX} was chosen as the score output format because it is powerful, flexible and well-documented, MusiX_{TEX} files can be compiled to PostScript files with freely available packages. Also, since MusiX_{TEX} files are text documents, the output scores can be edited and corrected with any text editor.

The motivation for implementing score generation as a component in a transcription system (as opposed to using an external program) is that the transcription system has vastly more information about the signal at various levels at its disposal than would an external commercial package which uses MIDI files only. However, score generation is a very difficult process which would entail (amongst many others):

- displaying chords correctly,
- inserting beams and slurs correctly,
- deciding on the best way to represent accidentals (for example, the same pitch can be written as C \sharp , B \times or D \flat depending on the context), and
- notating repetitions in the performance.

In our system, notes were notated individually³, using the correct pitch height and accidentals for each note, according to the key signature as extracted from the music. Since the system does not perform an analysis of the meter, and thus does not extract information about the beat or the measure, coherent bar divisions could not be done. The system groups every six notes into one bar to make the output more readable. Due to limitations of the system, score generation could not be standardised further than this. This attempt at generating score output was nonetheless an exercise that proved how much work still remains to be done even at such a high level where most of the processing is generally considered to be finished already!

6.7 Comments based on the synthetic signal

A piano roll excerpt plotted from the MIDI output for the synthetic signal is given in Figure 6.3. All component notes of the original signal were correctly identified, and there are no insertions or deletions. The synthetic signal had some very special properties which are not universally applicable to real signals:

- It contained no noise⁴.
- All partials were perfectly harmonic.
- All component notes were equally loud.
- Each note was composed of a limited number of harmonics only, of which the fundamental dominated.

The synthetic mixture did however conform with the generally accepted model of polyphonic music signals as the sum of concurrent sinusoids (*cf.* Equation 5.3). The fact that many of the algorithms gave correct results for synthetic signals indicates that the basic underlying concepts were correctly identified and implemented.

A last important comment needs to be made so as to emphasise the significance of the success of the system with synthetic signals: Synthetic signals were not successfully transcribed because they were *synthetic*; instead, they were successfully transcribed because they conformed with some *model* of instrument sounds. As has been pointed out several times during the course of this work, the lack of instrument models is the single most important reason for the relatively mediocre performance of the current multi-pitch estimator when used with real signals. It is imperative that future refinements to the system make use of more accurate models in all aspects of processing.

³I.e. with no slur or beam formation

⁴Apart from an impulse at the beginning and end of each note resulting from the synthesis method.

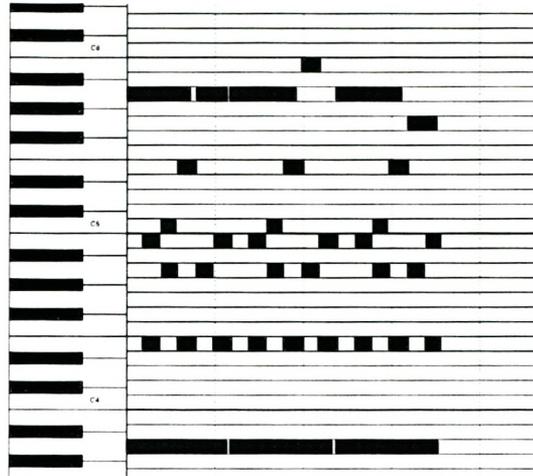


Figure 6.3: *Piano roll excerpt for the synthetic polyphonic sound*

After a brief discussion of the implementation of the system in the next chapter, transcription results obtained with the system are given in Chapter 8.

Chapter 7

Implementation issues

7.1 Development of a music processing library

Due to the lack of an established culture of music technology research at our university, most of the basic building blocks of the transcription system were implemented from scratch, excepting a WAV reader and an FFT implementation, both of which were taken from the PatRecII pattern recognition system developed at the University of Stellenbosch. During implementation, special care was taken to design the system components in such a way that they can later be used or integrated in other music analysis programs also. For the storage and processing of symbolic music data, a data hierarchy was formed, as shown in Figure 7.1.

7.2 Speed

Initial development was undertaken in the MATLAB environment, due to its large toolbox of signal processing functions and easy-to-use visualisation features. However, once promising solutions were found, MATLAB's execution speed proved a daunting hurdle

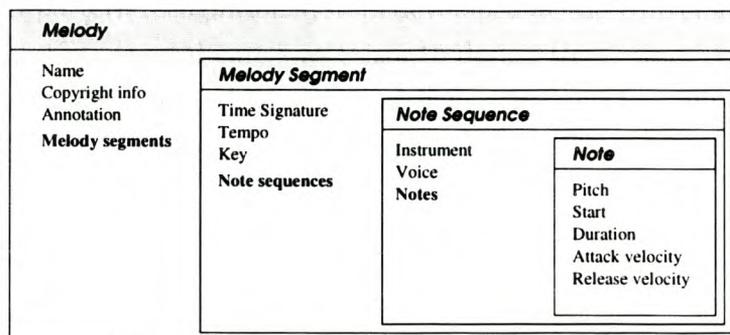


Figure 7.1: *Hierarchy of the symbolic music data in the system*

to quickly fine-tuning certain parameters or testing aspects of the system on different samples.

The transcription system was thus implemented in C++, where even with some awkward programming in certain parts of the system, the duration it takes to process a sample is of the same order as the actual duration of the sample. The slowest part of the transcription system is the multi-pitch estimation stage because the volume of data at this stage is very large, whilst the information density is fairly low. Each subsequent stage takes less processing time because as the information quality increases, the data quantity decreases, as shown in Figure 2.1.

The current implementation is not suited for real-time use due to the fact that all pitch tracking is completed first before moving on to the post-processing phase. This is done to allow the system to process information both forward and backward in time so that it needs less prior knowledge of the signal and has to make the least number of assumptions regarding the content.

7.3 User-friendliness

For a transcription system to be truly useful, it needs a user-friendly front-end, where segments of the signal can be visualised and processed, extracted information displayed, and the results examined and possibly edited. The current system is command-line based, with little or no user intervention. Output is done to formats which require external software. The MIDI files that the system produces can be listened to with any multimedia player with MIDI capabilities, and can also be edited by a number of commercially available packages. For the reasons explained in Section 6.6, scores are output to MusiX \TeX format, which can be edited with any text editor and for which there are freely available distributions on most computer platforms. A simple MATLAB program was also written to visualise MIDI files on piano rolls.

Neither the MIDI nor the MusiX \TeX output of the implementation is currently sufficiently formatted and musically accurate for comfortable use and general application. It is to be hoped that in future, a more user-friendly front-end for the program will be written which would make it easier to relate different aspects of the system (waveform, spectrogram, raw and processed pitch tracks, piano rolls, sound and notation) to each other.

Chapter 8

Experimental Investigation

8.1 Introduction

In the foregoing chapters, algorithms were described which were implemented as part of a transcription system. Several music files were transcribed and are presented here to give an impression of the different successes and failures of the transcription system. The results are given in increasing order of complexity, from simple monophonic transcriptions to drastically more involved polyphonic music transcriptions.

Because of the (relatively) simple nature of monophonic output data, objective accuracy measures can be calculated for them by comparing the acoustic input with the transcribed scores and piano rolls. Such measures will thus be given for the two monophonic transcriptions.

For polyphonic samples, the output is still so raw that valid objective measures can hardly be calculated efficiently. For example, the most common transcription mistakes are octave errors and deleted (missing) notes. Both of these types of errors are determined by the inherent design limitations of the system: as has been reiterated numerous times throughout the previous chapters, a system that does not incorporate extensive music knowledge sources, instrument models and advanced auditory cues cannot work reliably on general polyphonic signals. For that reason, the system was shown to work earlier with controlled synthetic polyphonic signals which met the constraints placed upon the algorithms. The polyphonic signals tested here serve to underline the described limitations of the system.

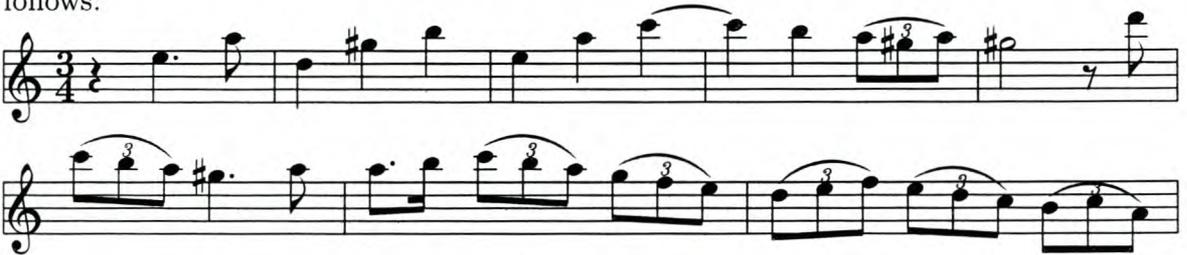
Because of their bulk which would necessarily distract from the flow of the following discussions, the visual representations of transcription results were relegated to Appendix G.

8.2 Transcription of a monophonic recorder sample

8.2.1 Results

Recorder and flute music are relatively simple to transcribe, and indeed, numerous transcription systems described in literature expect such samples [16]. The reason for this is the approximately sinusoidal nature of such signals: the fundamental of the signal is by far the most prominent partial by a margin of at least 10 dB, as seen in Figure G.1(a).

The chosen sample is the recorder part of G.F. Handel’s “*Sonata in a minor for Recorder and figured bass (Op. I, no. 4)*”, for which the first few bars are given as follows:



Results for various steps during the transcription process are given in Figure G.1. The final result is plotted as a piano roll in Figure G.2.

The corresponding bars of the M_usiX_TE_X transcription were generated as follows:



The 3’s under some notes indicate triplet values. The accuracy of the first 20 seconds of the transcription of the recorder sample is given in Table 8.1. It is noteworthy that the system did not make a single transcription mistake. Although there are notes which do not appear in the original score, these all formed part of the recording in the form of ornamentation or performance mistakes.

Table 8.1: Accuracy of the recorder transcription

Original notes	Correct	Insertions	Deletions	Substitutions
42	42	0	0	0

8.2.2 Discussion

The recorder transcription is remarkably accurate due to the “ideal” spectrum of recorder sounds. The sample was played with some insecurity, which gave it a somewhat unsteady beat and led to a number of minor erroneous notes. Any resulting deviations of the transcription from the original score are thus attributable to *performance inaccuracies*, one of the two sources of errors described in Section 4.1.4. The sample contained fairly loud breathing noises, all of which were removed by the post-processor.

Monophonic music transcription is often viewed disparagingly as a “glorified” version of straightforward pitch tracking. That is most definitely not an accurate assessment: the higher-level processing components required for generating notated scores for monophonic signals are far from trivial. Even though recorder samples are hardly representative of general music samples due to their simple spectral characteristics, the results of monophonic transcription summarised here mark a very positive first step in the evolution of a comprehensive transcription system.

8.3 Transcription of a monophonic violin sample

8.3.1 Results

Spectrally less ideal than the recorder is the violin; as can be seen from Figure G.3(a) the violin’s fundamental is typically not the strongest partial in its spectrum. Apart from a more complex spectrum, the tested violin sample was also significantly faster than the Handel recorder sample, with a wider range of notes played. Furthermore, it can be seen in the spectrogram of the sample in Figure G.3(c) that the fundamental and second partials of the sound often disappear. The first few bars of the original score of *Partita No. 2: Allemanda* by J.S. Bach are given as follows:



The results of various stages of the transcription process are provided in Figure G.3 together with a corresponding piano roll in Figure G.4.

The transcription program generated the following output for the first few bars of the violin sample:

♩ = 70

An overview of the accuracy of the first 20 seconds of the transcription is given in Table 8.2.

8.3.2 Discussion

As can be seen from Table 8.2, the violin transcription is less accurate than the recorder transcription. This is mostly due to the fact that the harmonic structure of violin sounds is much more complex, with very strong upper partials. Because of this, the transcription is prone to octave errors: all of the substitutions in Table 8.2 are octave and other harmonic errors. The system is especially prone to transcribing notes lower than Middle C an octave higher. This is probably a result of the fact that on most instruments, the fundamental becomes relatively weaker as the pitch is lowered. Apart from the harmonic errors, a number of notes appear in the transcription that were not part of the original score. However, these can all be explained as either performance inaccuracies or incidental sounds (such as the bow scraping on adjacent strings) generated at note transitions and are thus not truly insertions.

In spite of the few small mistakes, the transcription is still a very precise rendition of the performance, and a good approximation of the original score. As with the recorder sample, the key was detected correctly. The note duration symbols are not rhythmically coherent, for the reasons that were explained in Section 6.5.3. Nevertheless, they do give

Table 8.2: *Accuracy of the violin transcription*

<i>Original notes</i>	<i>Correct</i>	<i>Insertions</i>	<i>Deletions</i>	<i>Substitutions</i>
80	73	0	0	7

an accurate impression of the relative note durations in the context of the performance.

8.4 Transcription of a polyphonic organ sample

8.4.1 Results

The “*Tocatta in D Minor (BWV. 565)*” by J.S. Bach was chosen as an organ test sample. From Figure G.5(a) it can be seen that the organ spectrum has strongly sinusoidal partials. Organ recordings could thus be assumed to be fairly convenient polyphonic music to transcribe. However, as can be seen and heard from the results (a piano roll is given in Figure G.6), this is not necessarily the case.

8.4.2 Discussion

In the transcription of the Bach sample it becomes apparent that although a large fraction of the notes are in fact detected, the majority of them do not stay in one fixed octave register. The effect of this when listening to the MIDI transcription is disconcerting: although the melody can be identified, the constant octave changes (coupled with occasional incorrect notes) cause a constant maelstrom of sounds which obscures the flow of the piece to a large extent.

Organ sounds have incredibly rich spectra which cause their bombastic and majestic timbre. The unfortunate side effect for “untrained” music transcription systems is that when a number of notes are played together, their spectra overlap and fuse to create pseudo-notes which can “fool” the heuristics of the current multi-pitch estimation algorithm into validating them incorrectly as true notes.

This effect will need to be overcome in future enhancements.

8.5 Transcription of a polyphonic piano sample

8.5.1 Results

The track “*The Heart Asks Pleasure First*” by Michael Nyman (from the soundtrack to the movie *The Piano*) was used to demonstrate the transcription of an non-synthetic, acoustic piano sample. Note that the score of this composition was used for generating the synthetic test signal in Chapters 5 and 6. A piano roll of the transcription is shown in Figure G.8.

8.5.2 Discussion

Whilst the synthetic version of this composition was analysed flawlessly, the genuine piano performance was transcribed only very roughly. Notes often break up erratically; octaves are detected incorrectly; many notes are missing. The reasons for this include:

- Real piano sounds have prominent attacks, after which they slowly decay. In the current system, given its limited pitch tracking abilities (due to the lack of a metric pulse analysis front-end), decaying notes are often “lost” amidst the general spectral mayhem generated by new note attacks.
- The sample was played with generous use of the sostenuto pedal, and thus all notes from one measure decay whilst new ones are continuously added, creating a very dense, fairly flat spectrum. This effect can also be seen in the very “white” spectrogram of the sample in Figure G.7(c).
- The sample was played very fast, and thus new notes are introduced at an alarmingly quick rate. The transcription system is thus faced with a dense spectrum that it cannot handle effectively. This implies that the stream tracing mechanism depicted in Figure 3.3 should be implemented to subtract detected streams from the mixture so that only new streams have to be analysed. This would remove the burden of having to detect all component pitches (including those detected before) in every frame. Instead, with “old” pitches subtracted, only new pitches need to be detected in a much cleaner spectrum. This idea will be revisited in the recommendations for further research in the concluding chapter.

8.6 Observations

The results discussed above lead to the following general observations:

- Monophonic music transcription was achieved successfully within the limited scope of the project. Refinements can be made to format the output and to eliminate the few errors that do occur, but all-in-all the system should not require major re-designing in future.
- Polyphonic music transcription was only achieved successfully for synthetic signals (which were transcribed flawlessly). For non-synthetic signals, the number of transcription errors exceeds the number of successfully detected true notes. Octave errors (which qualify as substitutions) are the main cause of problems, though deletions (often arising from merged spectra which the system failed to separate) are also common. If octave errors were disregarded, and only the detected pitch classes

examined, the system actually performs well¹. This is also the main reason why the system managed to correctly identify the keys for a large number of tested music signals.

- The samples which were used were not chosen to play into the strengths of the system. In fact, signals that “break” a system have much greater instructional value in that they point out weaknesses and deficiencies. In that sense, the mediocre performance of the system on polyphonic samples serves to indicate areas of future research as given in the conclusion.
- For monophonic samples, the main cause of errors are imperfections in the performance (*i.e. human errors*). For polyphonic music, errors arise mostly from erroneous processing decisions stemming from the lack of sufficient knowledge sources (*i.e. machine errors*).
- Because of the lack of labelled polyphonic data, the discussions for the tested polyphonic signals were necessarily based on *subjective* measures. Once an appropriate sequencer with a large sample bank is obtained, the system should be tested more exhaustively with MIDI synthesised signals, which would allow for direct calculation of *objective* accuracy measures.

In conclusion it can thus be said that the system *does* work, as proven by the synthetic signal, but only under very controlled, limited conditions. Overcoming these will be the main challenge for further work.

¹Many published transcription systems in fact only detect pitch classes. Martin’s first blackboard system formed chords without knowledge of the precise octaves involved.

Chapter 9

Conclusions and Recommendations

9.1 The story thus far

Over the foregoing chapters, the AMADEUS music analysis and transcription system was developed based on some of the more important theories of human audition, musical acoustics and music theory. Exploring the field of automatic music transcription is daunting, and the familiarity with the subject matter gained throughout the course of the project has only deepened our impression that the implementation of a complete and successful automatic transcription system is a vast undertaking. Initial optimism quickly gave way to consternation as the scope of the problem became apparent! The following seem to be the key reasons which make the automatic transcription of polyphonic music so complex:

- *Reverse-engineering of a non-invertible operation:* The superposition of sound waveforms is a non-invertible operation. Even worse: music is often composed in such a way that the generating sources blend together even more completely to fool the human auditory system so that individual physical sources cannot be identified.
- *The interdependence of solutions to the various sub-problems:* None of the sub-problems of automatic music transcription can be seen in isolation. The results of each processing component need to be integrated with the results from virtually every other processing component in order to provide accurate results. This suggests that extensive horizontal, bottom-up and top-down processing structures need to be implemented in order to solve the problem.
- *The human factor:* Music is an art form that strives to express ideas and emotions – human traits that are not well suited for machine processing. As with all expressions of art, there are many subtle and less subtle ambiguities in music; the tomes that have been written throughout history advocating often wildly divergent interpretations of certain music pieces attest to this! These ambiguities in composition and performance add to the allure of music, yet they also beg the question: if humans

themselves (being the target audience) are not given to agreement on the correct analysis of a music piece, how can a machine fare better?

Given this complexity of the problem, the outcomes of a transcription system need to be better defined. It is hardly realistic to define automatic music transcription as the process whereby the *original score* is reconstructed from a *single* performance instance. Even Mozart's transcription of Allegri's *Miserere* was said to have contained the performance improvisations typical of the era. Automatic music transcription is better defined by requiring the following outcomes:

- The score output of the program should be a valid, reasonable representation of the recorded performance which was analysed.
- The system should eliminate those features which are obviously expressive performance characteristics to allow for less cumbersome output, without over-simplifying the content. In other words, the system should “filter” the output to make the musical content clearer than a raw transcription would be, without losing the nuances of the original composition.
- Features that remain strictly constant irrespective of the particular performance should be accurately detected. Such features include tonality and key signature.

Given the above refined requirements, it is time to take stock of the status of our transcription system. The many inherent limitations of the system have been detailed throughout the development in Chapters 4 to 6 and will not be repeated here. These limitations stem from the fact that the system is not intended as the be-all and end-all of automatic music transcription. Instead, the development of the system served as an exploratory study of the field of automatic music transcription and analysis.

The following comments can be made about the system and its performance:

- The system contains components from all stages of processing. Very few transcription systems have been designed that even attempt to run the full course of transcription. Most research focuses on individual aspect of transcription only. As such, the scope of the project (given the limited time frame and scarce resources) was very ambitious¹. However, the results do indicate that the system gives results which, inspite of being deeply flawed in the general polyphonic case, are nonetheless recognisable renditions of the original signal because they generally correctly identify *pitch classes* as opposed to the correct *pitches*.
- It is to be kept in mind that all the samples discussed in the previous chapter were real signals. Many algorithms in literature were tested mainly with fairly well-

¹The saying “Fools rush in where angels fear to tread” comes to mind.

behaved synthetic signals by their authors. Such tests allow for a more controlled and quantifiable evaluation of various algorithms. However, we felt that more insight into the nature of music and the difficulties of transcription could be obtained by focussing on real signals, even though the output generated from them may be less than ideal.

- Monophonic music can be transcribed by AMADEUS to a large degree of accuracy, even generating raw scores with standard music symbols (though without musical phrasing). The results of monophonic transcription are good enough that people who are musically literate would be able to use them in conjunction with the original audio signal to recreate accurate scores of the original performance.
- The system struggles with multi-pitch extraction from actual polyphonic music signals, as can realistically be expected. The multi-pitch estimation algorithm itself would not appear to be the weak link in the system, as tests with synthetic signals succeeded, suggesting that the underlying mechanisms of the algorithm are correct. The problem lies in the combined facts that no accurate signal models are used and that the knowledge sources of the system are extremely limited.
- The key detection algorithm works correctly, even with the flawed output of the polyphonic pitch tracker. The implications of this are two-fold: Firstly, this indicates that the pitch tracker does in fact provide much accurate data about the signal. Secondly, the fact that some higher level data can be extracted accurately from the raw transcription suggests that top-down processing structures feeding into the pitch estimators and trackers may greatly improve the transcription accuracy.

9.2 The road ahead

From our conclusions above, it can be seen that even though the current system is an important first step towards a fully functional system, a long and winding road lies yet ahead. Following are the areas of research which we propose for future improvements to the AMADEUS system:

- It is imperative that extensive knowledge sources be formulated and incorporated at all levels of the system. First and foremost of these is the development of instrument timbre models which form a crucial knowledge source for multi-pitch extraction. Another outcome of timbre modelling is instrument recognition, a very useful processing tool in its own right.
- The second deficiency of the system that urgently needs to be addressed is the current lack of a beat tracking component. The state-of-the-art beat tracking algo-

rithms have been described in Chapter 2. These should be used as a starting point for future expansions of AMADEUS.

- Fourier-based analysis, as used in the current pitch estimation algorithm, works fine for resolving partials of frequencies above 1 kHz where the true resolution is finer than a quarter tone when 46.4ms frames are used. However, for polyphonic music with mixtures of many simultaneous notes below Middle C, the poor frequency resolution becomes a real problem in resolving the individual signals. Furthermore, the fact that the FFT bins are spaced linearly in frequency is non-ideal for music. For these reasons, other multi-pitch extraction techniques should be investigated:
 - As discussed in earlier chapters, correlation-based pitch trackers have been intensively researched over the past half-decade, and a number of published results indicate that such techniques may produce the most viable multi-pitch estimation method. Our future research should experimentally investigate these methods.
 - An early implementation of our monophonic multi-pitch estimation component used the Constant Q transform. Constant Q signals were correlated with a harmonic model, and the largest correlation value indicated the pitch. This method is a very elegant solution to pitch detection, but our preliminary results suggested that this method is not very robust for signals with even low levels of noise. However, an interesting path of research would be to investigate the possibility of using a database of chord models (instead of single notes) as the data with which the signal is correlated. The best-matching chord for each analysis frame can be determined from the set of correlation results². Furthermore, calculating the derivatives of Constant Q values between adjacent frames would generate large positive values at all frequencies which correspond with newly-initiated signal streams, and large negative values for all frequencies that ended simultaneously. An algorithm based on this would incorporate Bregman’s auditory cues of *common onset*, *harmonicity* and *common amplitude variation*, as well as the method he described whereby humans trace existing streams³. It can also be noted that removing already detected notes from the mixture is a great deal simpler for Constant Q representations than for typical FFT spectra because of the typically greater frequency width of the Constant Q bins. The bell shape and side lobes of sinusoidal components complicate

²This is in keeping with research that suggests that humans recognise chords instead of individual notes [20].

³This would be an expansion of the work of Hawley, as described by Martin in [38].

spectral subtraction in FFT spectra.

- Human audition is explained by two different theories: *place* theory and *periodicity* theory⁴. Modern theories of hearing venture that a combination of the methods is used by the auditory system. Thus it would be worthwhile to investigate a polyphonic pitch tracker which calculates several multi-pitch estimates from both the time- and the frequency-domains, and combines these in some fashion⁵. Such a scheme would compensate for the deficiencies of one representation by complementing it with the strengths of another. Given the tremendous advances in computer processing power over the past few years, such a computationally expensive algorithm appears considerably less absurd today than it would a decade ago.
- Of all the aspects of automatic music transcription, the use of higher-level musicological models has been the least researched and published. Such models would include chord, chord transition and rhythmic knowledge. A variety of pattern recognition methods such as Neural Networks, Hidden Markov Models and Bayesian Probability Networks would be worthy of investigation in applying the models to the data. Conversely, an interesting application of a transcription system would be to automatically extract these models from transcribed data, creating a truly powerful automatic music *analysis* tool.
- Encompassing all of the foregoing recommendations, a top-down framework should be implemented to allow for the flow of information both *up* and *down* the data abstraction and processing hierarchies.

The field of automatic music transcription thus remains wide-open with vast opportunities for original research. The journey has only just begun...

*Transcription is a difficult thing
Finding every note from every string
Rhythm, beat and key
Notated for all to see
Oh what joy doth music bring!*

⁴As explained in Chapter 3.

⁵Such a combined estimate has precedent in monophonic pitch tracking with the Gold-Rabiner method which uses six independent pitch estimators [41].

Bibliography

- [1] ASKILL, J., *Physics of Musical Sound*. New York: D. Van Nostrand Company, 1979.
- [2] BACH, J., “Complete organ works, vol. ii.” Sheet music.
- [3] BREGMAN, A., *Auditory Scene Analysis*. Cambridge, Massachusetts: The MIT Press, 1990.
- [4] BROWN, J., “Calculation of a constant Q transform.” *Journal of the Acoustical Society of America*, January 1991, Vol. 89, No. 1, pp. 425–434.
- [5] BROWN, J. and PUCKETTE, M., “An efficient algorithm for the calculation of a constant Q transform.” *Journal of the Acoustical Society of America*, April 1992, Vol. 92, No. 5, pp. 2698–2701.
- [6] CEMGIL, A., “Automated Monophonic Music Transcription: A Wavelet Theoretical Approach.” Master’s thesis, Boğaziçi University, 1995.
- [7] CEMGIL, A., DESAIN, P., and KAPPEN, B., “Rhythm Quantization for Transcription.” in *Proceedings of the AISB’99 Symposium on Musical Creativity*, pp. 64–69, 1999.
- [8] CEMGIL, A., KAPPEN, B., DESAIN, P., and HONING, H., “On tempo tracking: Tempogram Representation and Kalman Filtering.” *Journal of New Music Research*, 29 (4), 2000, pp. 259–273.
- [9] CHIEN, Y.-R. and JENG, S.-K., “An Automatic Transcription System with Octave Detection.” *Proceedings of IEEE International Conference on Acoustic, Speech, and Signal Processing*, 2002.
- [10] CLASSICAL.NET, “Gregorio allegri: Miserere.” World-Wide Web: <http://www.classical.net/music/comp.lst/works/allegri/miserere.html>.
- [11] COGAN, R. and ESCOT, P., *Sonic Design: The Nature of Sound and Music*. New York: Prentice-Hall, Inc., 1976.

- [12] COOKE, P., "Heterophony." in *The New Grove Dictionary of Music and Musicians* (SADIE, S. (Ed.)), London: Macmillan Publishers Limited, Second edition, 2001.
- [13] CORBETT, J. and YELVERTON, V., *Music for G.C.E. 'O' Level*. London: Barrie and Rockliff, 1961.
- [14] CULVER, C., *Musical Acoustics*. Third edition. Philadelphia: The Blakiston Company, 1951.
- [15] DIXON, S., "A Beat Tracking System for Audio Signals." *Proceedings of the Conference on Mathematical and Computational Methods in Music, Vienna, Austria, December 1999*, pp 101-110, 2000.
- [16] FERNÁNDEZ, P. and CASAJÚS-QUIRÓS, F., "Multi-pitch estimation for polyphonic musical signals." *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing*, 1998.
- [17] FITCH, J. and SHABANA, W., "A Wavelet-based pitch detector for musical signals." *2nd COST-G6 Workshop on Digital Audio Effects, December 1999*, 1999.
- [18] FORTUNE, N. and CARTER, T., "Monody." in *The New Grove Dictionary of Music and Musicians* (SADIE, S. (Ed.)), London: Macmillan Publishers Limited, Second edition, 2001.
- [19] FROBENIUS, W., "Polyphony." in *The New Grove Dictionary of Music and Musicians* (SADIE, S. (Ed.)), London: Macmillan Publishers Limited, Second edition, 2001.
- [20] GERHARD, D., "Computer Music Analysis." *Technical Report CMPT TR 97-13, Simon Fraser University*, 1997.
- [21] HAMILTON, C., *Sound and Its Relation to Music*. Boston: Oliver Ditson Company, 1912.
- [22] HASTY, C., *Meter as Rhythm*. New York: Oxford University Press, 1997.
- [23] HYER, B., "Homophony." in *The New Grove Dictionary of Music and Musicians* (SADIE, S. (Ed.)), London: Macmillan Publishers Limited, Second edition, 2001.
- [24] JOSEPHS, J., *The Physics of Musical Sound*. Princeton, New Jersey: D. Van Nostrand Company, Inc., 1967.
- [25] KARJALAINEN, M. and TOLONEN, T., "Multi-pitch and Periodicity Analysis Model for Sound Separation and Auditory Scene Analysis." *Proceedings of IEEE International Conference on Acoustics Speech and Signal Processing*, 1999, Vol. 2, pp. 929-932.

- [26] KASHINO, K., NAKADAI, K., KINOSHITA, T., and TANAKA, H., "Application of Bayesian Probability Network to Music Scene Analysis." *Proceedings of the International Joint Conference on AI, CASA workshop*, 1995.
- [27] KASHINO, K., NAKADAI, K., KINOSHITA, T., and TANAKA, H., "Organization of Hierarchical Perceptual Sounds." *International Joint Conference on AI*, 1995, pp. 158–164.
- [28] KLAPURI, A., "Automatic Transcription of Music." Master's thesis, Tampere University of Technology, 1997.
- [29] KLAPURI, A., ERONEN, A., SEPPÄNEN, J., and VIRTANEN, T., "Automatic Transcription of Musical Recordings." *Consistent & Reliable Acoustic Cues Workshop, CRAC-01, Aalborg, Denmark*, September 2001.
- [30] KLAPURI, A., "Musical meter estimation and music transcription." Paper presented at the Cambridge Music Processing Colloquium, Cambridge University, UK, 2003.
- [31] KLINGSEISEN, J. and PLUMBLEY, M., "Experiments on musical instrument separation using multiple-cause models." *Proceedings of the Cambridge Music Processing Colloquium, Cambridge, England*, 30 Sept 1999.
- [32] KOSTKA, S. and PAYNE, D., *Tonal Harmony With An Introduction to Twentieth-Century Music*. Second edition. New York: McGraw-Hill, Inc., 1989.
- [33] KRUMHANSL, C., *Cognitive Foundations of Musical Pitch*. No. 17 in Oxford Psychology Series. New York: Oxford University Press, 1990.
- [34] LERDAHL, F. and JACKENDOFF, R., *A Generative Theory of Tonal Music*. Cambridge, Massachusetts: The MIT Press, 1990.
- [35] MAKHOUL, J., ROUCOS, S., and GISH, H., "Vector Quantization in Speech Coding." in *Proceedings of the IEEE*, vol. 73, pp. 1551–1588, November 1985.
- [36] MARQUARDT, A., TOERIEN, L., and TERBLANCHE, E., "Applying Nonlinear Smoothers to Remove Impulsive Noise from Experimentally Sampled Data." *N&O Joernaal*, April 1991, pp. 15–18.
- [37] MARTIN, K., "Automatic Transcription of Simple Polyphonic Music: Robust Front End Processing." in *M.I.T Media Laboratory Perceptual Computing Section Technical Report No. 399*.
- [38] MARTIN, K., "A Blackboard System for Automatic Music Transcription of Simple Polyphonic Music." in *M.I.T Media Laboratory Perceptual Computing Section Technical Report No. 385*.

- [39] MARTIN, K. and KIM, Y., “Musical instrument identification: A pattern-recognition approach.” Presented at the 136th meeting of the Acoustical Society of America, October 13, 1998.
- [40] MCNAB, R. and SMIGHT, L., “Evaluation of a Melody Transcription System.” *IEEE International Conference on Multimedia and Expo (II) 2000*, 2000, pp. 819–822.
- [41] MCNAB, R., SMITH, L., and WITTEN, I., “Signal Processing for Melody Transcription.” in *Proceedings of the 19th Australasian Computer Science Conference, Melbourne, Australia, January 31-February 2 1996*, 1996.
- [42] MONTI, G. and SANDLER, M., “Monophonic Transcription with Autocorrelation.” in *Proceedings of the COST G-6 Conference on Digital Audio Effects (DAFX-00), Verona, Italy, December 7-9, 2000*, 2000.
- [43] MOULINES, E. and LAROCHE, J., “Non-parametric techniques for pitch-scale and time-scale modifications of speech.” *Speech Communication*, 1995, No. 16, No. 16, pp. 175–205.
- [44] NYMAN, M., “The piano: Original compositions for solo piano.” Sheet music.
- [45] OLSON, H., *Music, Physics and Engineering*. Second edition. New York: Dover Publications, Inc., 1967.
- [46] PIERCE, J., *The Science of Musical Sound*. Revised edition. New York: W.H. Freeman and Company, 1992.
- [47] PROAKIS, J. and MANOLAKIS, D., *Digital Signal Processing: Principles, Algorithms, and Applications*. Third edition. New Jersey: Prentice-Hall International, 1996.
- [48] REGENER, E., *Pitch Notation and Equal Temperament: A Formal Study*. No. 6 in Publications: Occasional Papers. Berkeley: University of California Press, 1973.
- [49] RIGDEN, J., *Physics and the Sound of Music*. New York: John Wiley & Sons, 1977.
- [50] ROEDERER, J., *Introduction to the Physics and Psychophysics of Music*. Second edition. New York: Springer-Verlag, 1979.
- [51] ROHWER, C., “Variation and LULU-Smoothing.” *Journal of the South African Mathematical Society*, 2002, Vol. 25, No. 2, No. 2, pp. 163–176.
- [52] ROHWER, C., “Multiresolution Analysis With Pulses.” *Proceedings of the International Conference on Approximation Theory*, 2003, pp. 165 – 186.

- [53] ROSSING, T., *The Science of Sound*. Reading, Massachusetts: Addison-Wesley Publishing Company, 1982.
- [54] ROTHSTEIN, J., *MIDI: A Comprehensive Introduction*. Second edition. Madison, Wisconsin: A-R Editions, Inc., 1995.
- [55] SCHEIRER, E., "Extracting Expressive Performance Information from Recorded Music." Master's thesis, Massachusetts Institute of Technology, 1995.
- [56] SHUTTLEWORTH, T. and WILSON, R., "The recognition of musical structures using neural networks." IJCAI-95 Workshop on Artificial Intelligence and Music.
- [57] TAUPIN, D., MITCHELL, R., and EGLER, A., *MusiXTEX: Using T_EX to write polyphonic or instrumental music*.
- [58] TOLONEN, T. and KARJALAINEN, M., "A Computationally Efficient Multipitch Analysis Model." *IEEE trans. Speech and Audio Process.*, 2000, Vol. 8, No. 6, No. 6, pp. 708–716.
- [59] VIRTANEN, T. and KLAPURI, A., "Separation of Harmonics Sounds Using Linear Models for the Overtone Series." in *ICASSP 2002*, 2002.
- [60] WALMSLEY, P., *Signal Separation of Musical Instruments: First Year Report*. PhD thesis, Cambridge University Engineering Department, 1997.
- [61] WALMSLEY, P., GODSILL, S., and RAYNER, P., "Bayesian Modelling of Harmonic Signals for Polyphonic Music Tracking." *Cambridge Music Processing Colloquium*, 30 September 1999.
- [62] ZATORRE, R. and PERETZ, I. (Eds), *The Biological Foundations of Music*, vol. 930 of *Annals of the New York Academy of Sciences*. New York, New York: The New York Academy of Sciences, 2001.

Appendix A

Pitch and Notation

A.1 Pitch conversions

Important pitch conversions are summarised again by the following equations:

$$f_M = 440 \times 2^{\frac{M-69}{12}} \quad (\text{A.1})$$

$$M = 12 \log_2 \frac{f_M}{440} + 69 \quad (\text{A.2})$$

$$f_c = 440 \times 2^{\frac{c-6900}{1200}} \quad (\text{A.3})$$

$$c = 1200 \log_2 \frac{f_c}{440} + 6900 \quad (\text{A.4})$$

$$c = 100M \quad (\text{A.5})$$

$$\Delta M = 12 \log_2 \frac{f_a}{f_b} \quad (\text{A.6})$$

$$\Delta c = 1200 \log_2 \frac{f_a}{f_b} \quad (\text{A.7})$$

A.2 Notation

Various aspects of music notation, nomenclature and conventions are shown in the figures and tables over the next pages. Figure A.1 shows the notation and frequencies for all notes on the piano keyboard, along with selected sounding ranges.

Table A.1 lists the ratios and relative pitches of notes for scales based on C, for various tuning systems. The Pythagorean scales create the largest possible number of perfect fourths and fifths, and can be derived by traversing the circle of fifths (see [53, p. 156] for more details).

Table A.2 lists the standard duration symbols for notes and rests, along with their names.

Table A.3 gives the beat and meter divisions for time signatures that are common in classical music.

Figure A.1: Keyboard with notation (MIDI), pitch (Hz), octave registers, staff notation and selected sounding ranges

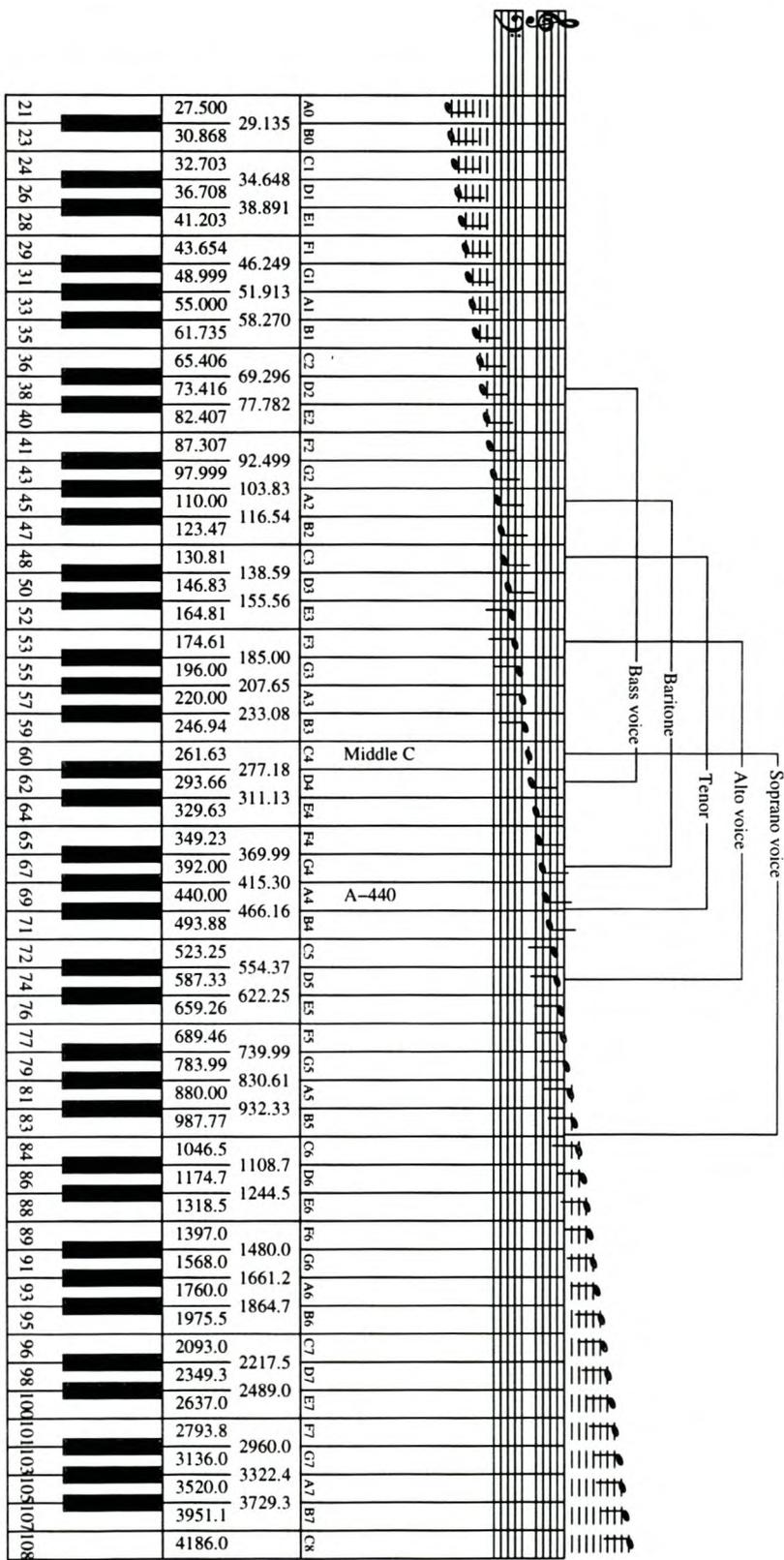


Table A.1: Notes of scales based on C (based on [53, Table 9.3])

Note	Tempered		Just		Pythagorean	
	Cents	Ratio	Cents	Ratio	Cents	
C	1200	2.000	1200	2.000	1200	
B♯	1200	1.953	1159	2.027	1224	
C♭	1100	1.920	1129	1.873	1086	
B	1100	1.875	1088	1.898	1110	
B♭	1000	1.800	1018	1.778	996	
A♯	1000	1.758	977	1.802	1020	
A	900	1.667	884	1.688	906	
A♭	800	1.600	814	1.580	792	
G♯	800	1.563	773	1.602	816	
G	700	1.500	702	1.500	702	
G♭	600	1.440	631	1.405	588	
F♯	600	1.406	590	1.424	612	
F	500	1.333	498	1.333	498	
F♯	500	1.302	457	1.352	522	
F♭	400	1.280	427	1.249	384	
E	400	1.250	368	1.266	408	
E♭	300	1.200	316	1.185	294	
D♯	300	1.172	275	1.201	318	
D	200	1.125	204	1.125	204	
D♭	100	1.067	112	1.054	90	
C♯	100	1.042	71	1.068	114	
C	0	1.000	0	1.000	0	

Table A.2: Duration symbols (based on [32, p. 26])

Value	Name	Note	Rest
Whole	Semibreve		
Half	Minim		
Quarter	Crotchet		
Eighth	Quaver		
Sixteenth	Semiquaver		
Thirty-second	Demisemiquaver		

Table A.3: *Typical time signatures*

<i>Time Signature</i>	<i>Beats per measure</i>	<i>Beat note</i>	<i>Division of the beat</i>
$\frac{2}{4}$	2		2× 
$\frac{2}{2}$	2		2× 
$\frac{3}{16}$	3		2× 
$\frac{3}{4}$	3		2× 
$\frac{4}{8}$	4		2× 
$\frac{4}{4}$	4		2× 
$\frac{6}{8}$	2		3× 
$\frac{6}{4}$	2		3× 
$\frac{9}{16}$	3		3× 
$\frac{9}{8}$	3		3× 
$\frac{12}{8}$	4		3× 
$\frac{12}{4}$	4		3× 

Appendix B

LULU Smoothing & Peak Detection Algorithm

B.1 Introduction

Peak detection is a common problem in DSP applications. A smoothing and peak detection algorithm that works well for FFT-based spectra is outlined briefly in this chapter. The technique uses non-linear *LULU*-smoothers developed by C.H. Rohwer of the University of Stellenbosch and is based on multiresolution analysis with pulses [52, 51].

The underlying principle of smoothing is *variation reduction*: the spread of values in the window under consideration must be reduced to an acceptable level. Linear smoothers replace each data point with some weighted average of points from its immediate neighborhood. A disadvantage to such a smoothing strategy is that linear filters smear large impulsive noise components across a number of points, instead of removing the noise. By contrast, non-linear filters generally replace each data point (which can contain a large additive noise component) with an acceptable value from its surroundings; non-linear smoothers are said to be *rank-based selectors* which select values from a window based on the relative ordering of the values in the window. The most popular non-linear smoother is the *median smoother* which replaces each signal point with the median of the points surrounding it. However, the median smoother is not idempotent, meaning that successive applications of the smoother to its own output can give different interpretations of the same original input data. This results in a degree of uncertainty or unpredictability in the smoother's behaviour [36].

Rohwer proposed a basic pair of non-linear smoothers, namely L and U , which can be used to overcome the limitations of the median smoothers. L removes upward outliers, whilst U removes downward outliers. In order to remove both up and down outliers, the

smoothers can be concatenated to form UL and LU smoothers¹. When designing these smoothers, the order n of the smoothers should be chosen as “at least the maximum number of consecutive expected outliers. A sequence of more than n “outliers” (in the same direction) is interpreted as a significant pattern in the data.” [36]

When filtering FFT spectra, the order n should be chosen so as to remove all side lobes, whilst leaving the main lobes intact. The main lobes of sinusoids are $R = \frac{N_{FFT}}{N_{win}}$ samples wide if square windows are used for the FFT. The main lobes are “significant patterns” in the data, the narrower side lobes are “outliers”. n should thus be chosen as $n = R - 1$ to remove the side lobes.

B.2 Algorithm

The following describes the basic algorithm for the smoother $U_n L_n$ [36]:

$$\begin{aligned} ULx(i) &= \min Z(i, i + n) \\ &= \min\{z(i), \dots, z(i + n)\}, \quad \text{where} \end{aligned} \quad (\text{B.1})$$

$$\begin{aligned} z(i) &= \max Y(i - n, i + n) \\ &= \max\{y(i - n), \dots, y(i), \dots, y(i + n)\}, \quad \text{with} \end{aligned} \quad (\text{B.2})$$

$$\begin{aligned} y(i) &= \min X(i - n, i) \\ &= \min\{x(i - n), \dots, x(i)\} \end{aligned} \quad (\text{B.3})$$

The smoother LU exchanges all minima and maxima in the above equations, and can be calculated as $LU(x) = -UL(-x)$. Marquardt *et al.* describe more advanced algorithms with which the above equations can be implemented efficiently using circular buffers [36].

Rohwer suggests that smoothing should be done successively by $U_1 L_1, U_2 L_2, \dots, U_n L_n$ to reduce the variation of the signal at each level of decomposition [51]. This was chosen as the method in which the smoother is implemented.

Peaks can be picked in a straight-forward manner from the spectrum by defining a peak to be a value that is greater than either of its neighbouring values. Provision has to be made for the fact that $LULU$ -smoothed signals generally have n repetitions of significant values, and peaks in such data are thus “peak plateaus”. The true peak is then found as the largest value in the original data in the range of indices of a $LULU$ peak plateau. Various elimination criteria based on thresholds can be devised to eliminate weak peaks.

¹Rohwer proves that UL smoothers provide the lower bound of median smoothed data, whilst LU smoothers provide an upper bound [51].

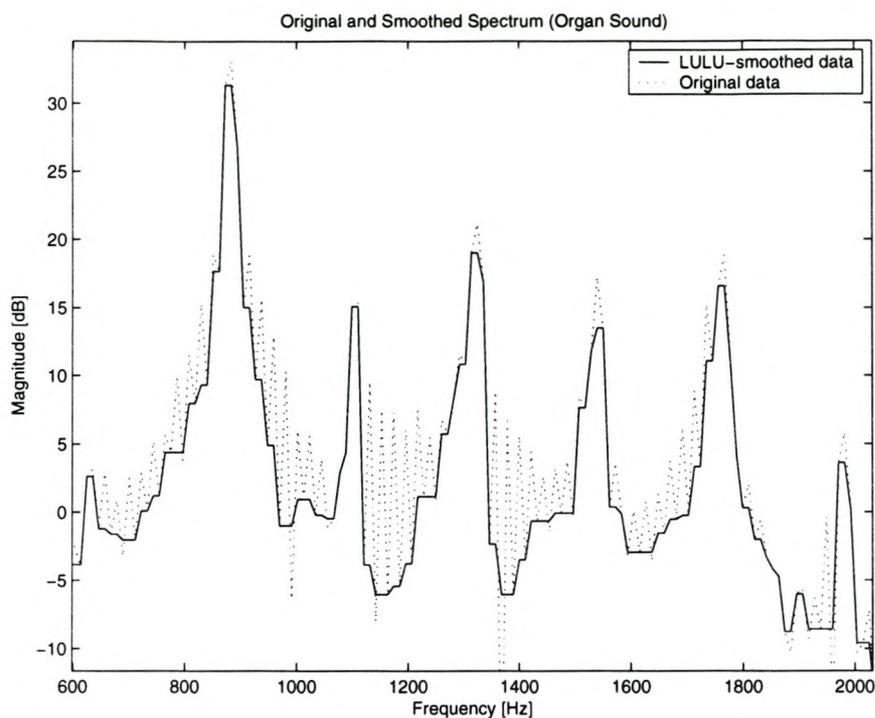


Figure B.1: Example of a LULU-filtered spectrum

B.3 Results

The above procedure is best illustrated with an example: Figure B.1 shows an organ spectrum smoothed with a *UL*-smoother. It can be seen that the smoother effectively removes all side lobes.

LULU-smoothers have found application in a wide variety of fields, including seismic data evaluation and image processing².

²An interesting image processing application is the use of *LULU*-smoothers to detect golf balls in photographs of golf courses.

Appendix C

Frequency Sharpening With The Phase Vocoder

C.1 Introduction

It is often desirable to find the instantaneous frequency of sinusoids in FFT spectra to a greater degree of accuracy than the time-frequency resolution allows. For that reason, a technique borrowed from the *phase vocoder* can be used to estimate the instantaneous frequency by making use of the phase changes of a certain FFT component from one frame to the next.

C.2 Algorithm

The calculation of the instantaneous frequency for a specific bin can be summarised as follows (for the detailed derivation, consult [43]):

1. Calculate the phase increment Z between two successive STFT spectra with N_{FFT} points and frame non-overlap lengths of N_{hop} samples, given the phases ϕ for both frames (frame 2 being the one further along in time) in FFT bin number N_{bin} :

$$Z = \phi_2 - \phi_1 - \frac{N_{bin}}{2\pi N_{FFT} N_{hop}} \quad (C.1)$$

2. Unwrap the phase increment Z to values between $-\pi$ and π , naming the result \bar{Z} .
3. The exact location $N_{bin,true}$ of the sinusoid in a bin can then be estimated using:

$$N_{bin,true} = N_{bin} + \frac{N_{FFT} \bar{Z}}{2\pi N_{hop}} \quad (C.2)$$

Note that whereas N_{bin} is an integer, $N_{bin,true}$ is a real number.

The most important constraints placed on the STFT analysis to allow for this method of frequency sharpening is that the *window must be longer than 4 pitch-periods* and that

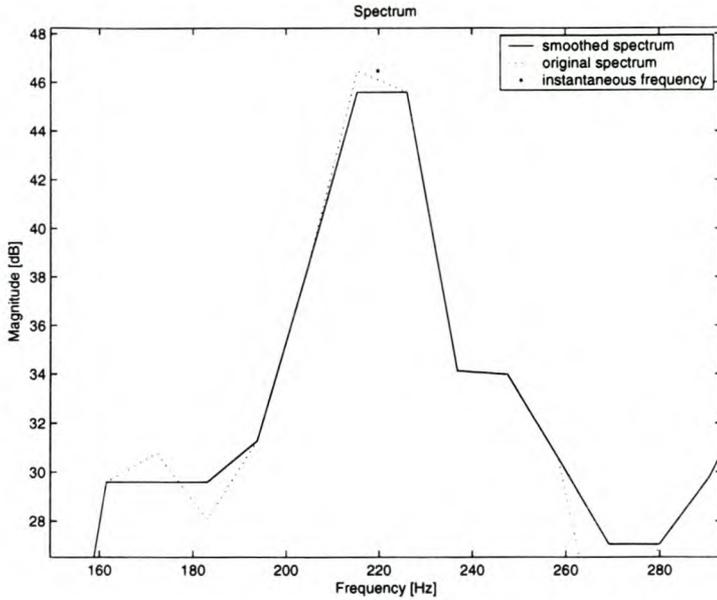


Figure C.1: *Instantaneous frequency of a sinusoid*

successive analysis windows must have a minimum overlap of 75% [43]. The latter of these constraints is satisfied in the multi-pitch estimation algorithm by letting the frames overlap by 87.5%. However, the former of these constraints is not necessarily satisfied, for two reasons: Firstly, with 46.4 ms frames, the lowest pitch which satisfies the constraint is 86 Hz (which is nearly an octave higher than the lowest note that typically needs to be detected). Secondly, for polyphonic signals, the pitch-period is not well-defined and thus the success of the sharpening cannot be guaranteed. However, because the sharpening technique is relatively computationally inexpensive, and because it is very successful for monophonic signals, it was left in the system when it was expanded from the monophonic to the polyphonic case. Moreover, the program makes use of partial candidates to increase the effective frequency resolution (as shown in Equation 5.17).

C.3 Results

A sharpened peak is shown in Figure C.1. The instantaneous frequency of the peak lies in-between two centre frequencies of FFT bins, as can be estimated from the bell shape of the peak. The peak position is accurately determined by the frequency sharpening algorithm.

As a demonstration of the practical use of instantaneous frequency determination, a monophonic pitch track for a scale sung by a male singer is given in Figure C.2. The actual resolution of the FFT for the analysis was around 20 Hz ($\frac{f_s}{N_{FFT}}$), yet as seen in

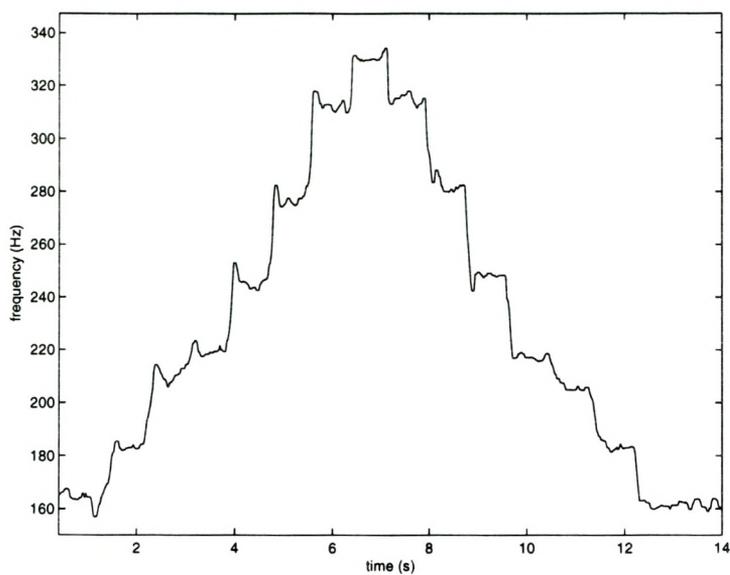


Figure C.2: *Pitch track of sung scale when sinusoidal frequencies are sharpened*

the figure, the pitch was ultimately resolved much closer than that: even the slightly sharp attacks on ascending notes and slightly flat attacks on descending notes (typical of singing) can be clearly identified.

Appendix D

Lloyd-Max Quantisation

D.1 Introduction

A number of algorithms in the current transcription system make use of scalar quantisation to group data or to find dynamic thresholds. The quantisation algorithm chosen for this is the well-known Lloyd-Max quantiser, which can be viewed as the one-dimensional case of the more general K -means algorithm which is commonly used for vector quantisation.

The quantiser finds the cell intervals C_i (with lower boundaries given by g_i) and corresponding interval centroids y_i which optimally fit the data to L levels (“bins”), as shown in Figure D.1. As in the more general case of K -means clustering, the intervals that are thus found may constitute a *local optimum* only, as opposed to the *global optimum*.

D.2 Algorithm

The Lloyd-Max scalar quantisation algorithm is given in [35] as follows:

$$g_i = \frac{1}{2}(y_i + y_{i-1}), \quad 2 \leq i \leq L \quad (\text{D.1})$$

$$\begin{aligned} g_1 &= -\infty, & g_{L+1} &= \infty \\ y_i &= \text{cent}(C_i), & 1 \leq i \leq L \end{aligned} \quad (\text{D.2})$$

where the centroid of C_i is simply the mean value of the data points in that interval.

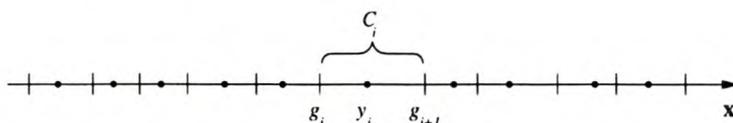


Figure D.1: Partitioning of the real line into cells (based on [35])

The equations are solved iteratively to obtain a set of optimal values, though these may be only local optima. Initial values are chosen such that all initial intervals C_i contain an equal number of data points (N/L , where N is the total number of data points).

Appendix E

The MIDI File Format

MIDI is the acronym for “Musical Instrument Digital Interface”, a data communications protocol that is used to exchange messages between music equipment, computers and software. It is a very broad and general standard which allows for the compact exchange of very nuanced data about musical performances. As such, MIDI provides the backbone of electronic music.

The specification of “Standard MIDI Files” was adopted as an addendum to the MIDI specification in 1988. The MIDI file format is the aspect of the MIDI specification which is most directly of concern in automatic music transcription, since transcription systems typically write their output to MIDI files. Exhaustive documentation of the MIDI file format is readily available on the Internet¹, as well as in countless books (for a good introduction, see [54]). Only the most important concepts will be outlined here.

MIDI files consist of one header chunk, which specifies the MIDI file format (of which there are currently three), the number of tracks, and a value for the division (which often, though not necessarily, gives the number of delta-time units per quarter-note). After the header chunk follows one or more track chunks.

Each track is encoded in a track chunk, which (apart from some house-keeping data) consists of a list of {Delta-Time, Track Event} pairs. The delta-time value specifies the amount of time (in MIDI clock ticks) which has elapsed since the previous track event. Track events can be *MIDI events*, *system exclusive events*, or *meta-events*. Meta-events provide “meta data” such as tempo, key signature, time signature, lyrics, instrument names and copyright information about the music. The MIDI events which are of most interest to automatic music transcription systems are the “Note on” and “Note off” messages. These messages encode the MIDI key, strike/release velocity and MIDI channel for each note event. These values correspond broadly with *pitch*, *loudness* and *instrument*, respectively. A corresponding “note off” message should be generated for all “note on”

¹For example: http://crystal.apana.org.au/ghansper/midi_introduction/contents.html

messages.

To put the compactness of the MIDI representation in perspective: “Note on/off” messages are encoded with three bytes in addition to the variable-length delta-time value (typically three or four bytes). Thus each note can be represented in MIDI files by only a dozen or so *bytes*, as opposed to the dozens of *kilobytes* of each note in a studio-quality sampled waveform. Given a good sample bank, extremely natural-sounding performances can be reconstructed from MIDI files.

Because notes are encoded in terms of {“Note on”, “Note off”} message pairs, MIDI files are not well-suited as a format for storing musical notation. Although the meta-events can be used to derive some notational information for each note, notated scores are generally stored in more suitable formats, one of which is described in Appendix F.

Appendix F

The MusiX_{TEX} Format

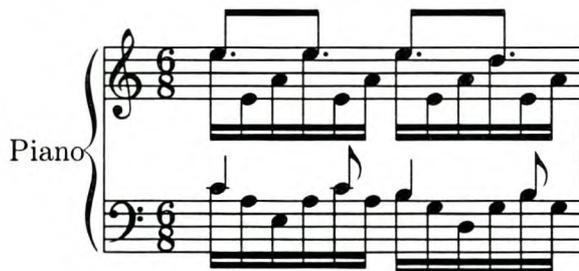
If you are not familiar with $\text{T}_{\text{E}}\text{X}$ at all I would recommend to find another software package to do musical typesetting. Setting up $\text{T}_{\text{E}}\text{X}$ and MusiX $\text{T}_{\text{E}}\text{X}$ on your machine and mastering it is an awesome job which gobbles up a lot of your time and disk space. But, once you master it...

HANS HUYKENS [57]

The MusiX $\text{T}_{\text{E}}\text{X}$ format is an extremely powerful set of $\text{T}_{\text{E}}\text{X}$ macros to typeset polyphonic music. The MusiX $\text{T}_{\text{E}}\text{X}$ manual [57] provides in-depth information about all facets of the package; this discussion serves merely to illustrate some of the aspects of its use.

MusiX $\text{T}_{\text{E}}\text{X}$ requires that scores be set-up before notes are entered: the number of instruments, the number of staves for each instrument, the clefs for each of these staves, the general key signature and the general time signature need to be specified. Notes are then entered in columns of *groups of simultaneous notes*, by using appropriate macros for note duration and pitch. Refer to Figure F.1 for an overview of the appropriate macros.

Score input is best illustrated with a short example. One bar from the score for Michael Nyman's "*The Heart Asks Pleasure First*" is given as follows:



The MusiX $\text{T}_{\text{E}}\text{X}$ sequence which was used to generate this is as follows:

```
\begin{music}  
\parindent10mm  
\instrumentnumber{1}
```

```

\setstaves 1{2}
\setname1{Piano}
\interstaff{12}
\setclef1\bass
\generalsignature{0}
\generalmeter{\meterfrac68}
\startextract
\notes\zqu c\ibbl0M0\qb0{caLa}\zcu c\qb0c\tbl0\qb0a%
\nextstaff\ibu110\zqbp1 l\ibbl0g0\qb0{leh}\tbu1\zqbp1 l\qb0{le}\tbl0\qb0h%
\notes
\notes\zqu b\ibbl0L0\qb0{bNKN}\zcu b\qb0b\tbl0\qb0N%
\nextstaff\ibu110\zqbp1 l\ibbl0g0\qb0{leh}\tbu1\zqbp1 k\qb0{ke}\tbl0\qb0h%
\notes
\endextract
\end{music}

```

Although this seems prohibitively complex at first, the commands themselves are very straight-forward (for example, `\ibbu...` initiates a double upper beam for demisemiquaver groups, whilst `\tbu...` terminates the beam). The fact that the commands are very simple yet powerful makes MusiX_TE_X an ideal tool for automated score generation¹. Also, MusiX_TE_X is not restricted to using standard modern notation: a large number of packages have been developed for typesetting anything from percussive music to Gregorian chant to guitar chords. This versatility also contributes to its usefulness as a score output format.

A MusiX_TE_X reference chart is given in Figure F.1.

¹All musical extracts in this thesis were prepared by hand using MusiX_TE_X. However, it should be kept in mind that MusiX_TE_X is discussed here in the context of automatic music transcription, and *not* in terms of its potential as a stand-alone typesetting package. As suggested by the introductory quote above, this method of musical typesetting requires a great deal of patience and experience before it produces rewarding results, and thus it is not necessarily suitable for the painstaking manual generation of complex scores.

MusiXTEX—Reference T.80

Pitches

'A 'B 'C 'D 'E 'F 'G A B C D E F G H I J K L M N a b c d e
 'a 'b 'c 'd 'e 'f 'g
 'A 'B 'C 'D 'E 'F 'G "A "B "C "D "E
 a b c d e f g h i j k l m n o p q r s t u v w x y z

Notes, Accidentals, Accents, Clefs and Rests

\zlonga \zmaxima \zbreve \zwmq \wh \hu \hl \qu \ql \cu \cl \ccu \ccl \cccu \cccl \ccccu \ccccl \grcu \grcl
 Accidentals: \z \cdsh \csh \cna \cfl i \cdf
 \dqu¹²³ \yqu¹²³ \dcqu² \dhqu² \doqu² \xqu² \oxqu² \roqu² \tgqu² \kqu² \squ³ \lsqu³ \rsqu³ \cqu⁴ \ql⁴ \chu⁴ \chl⁴
 1 musixdia.tex 2 musixper.tex 3 musixgre.tex 4 musixlit.tex 5 musixext.tex
 \lpz \upz \lsf \usf \lst \ust \lppz \uppz \lsfz \usfz \lpzst \upzst \downbow \upbow \flageolett \whp \qupp
 Accent on beam with prefix *b* and beamrefnumber instead pitch
 \trebleclef \bassclef \smallaltoclef \drumclef² \gregorianFclef³
 \altoclef \smalltrebleclef \smallbassclef \gregorianCclef³ \oldGclef⁴
 \qqs \hs \qs \ds \qp \hpause \hpausep \pause \pausep \PAuse \PAUse \Hpause⁴
 \liftpause

Other Symbols

\Trille \trille \shake \Shake \mordent \Mordent \turn \backturn \Shakel \Shakesw \Shakene \Shakenw
 tr
 \meterfrac \meterplus \allabreve \reverseallabreve \meterC \reverseC \duevolte \lpar \rpar
 \setvoltabox \setvolta \coda \Coda \segno \Segno \caesura \cbreath
 \metron 1. 2. \emptyset Π \S ped \PED \sPED \DEP \sDEP
 \fermataup \arpeggio d5 \uptrio \octfinup \slide⁵ \boxit A \circleit B
 \fermatadown \Fermataup \bracket \downtrio \octfindown \leftrepeat \leftrightrepeat \rightrepeat

Figure F.1: MusiXTEX Reference Chart (taken from [57])

Appendix G

Experimental Results

Results from the experimental investigation are given in the following figures, which describe:

- A monophonic recorder transcription
- A monophonic violin transcription
- A polyphonic organ transcription
- A polyphonic piano transcription

Sheet music extracts for the monophonic samples are given in Chapter 8. For the sheet music for the Bach and Nyman samples, consult [2] and [44] respectively.

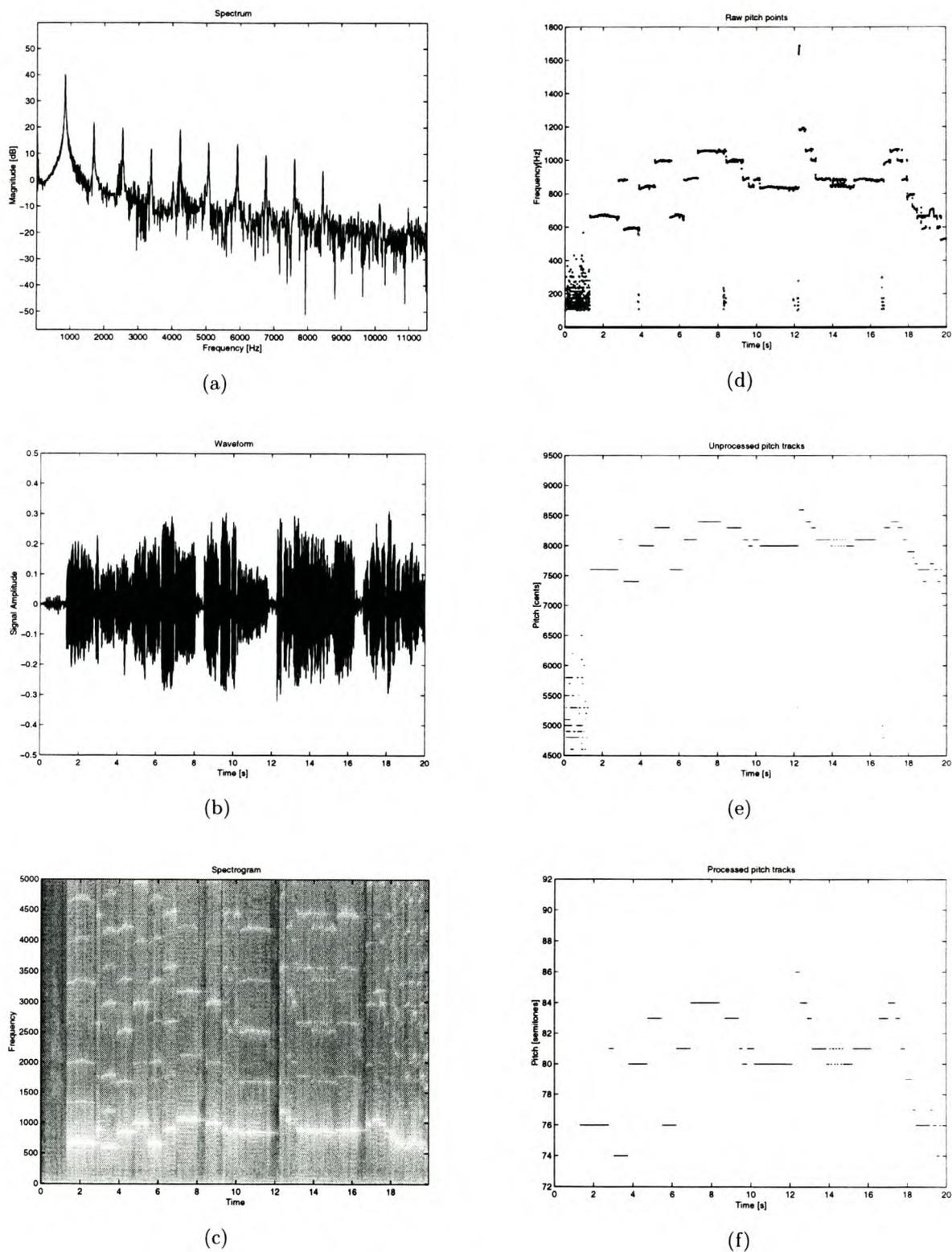


Figure G.1: Steps during the transcription of a recorder sample

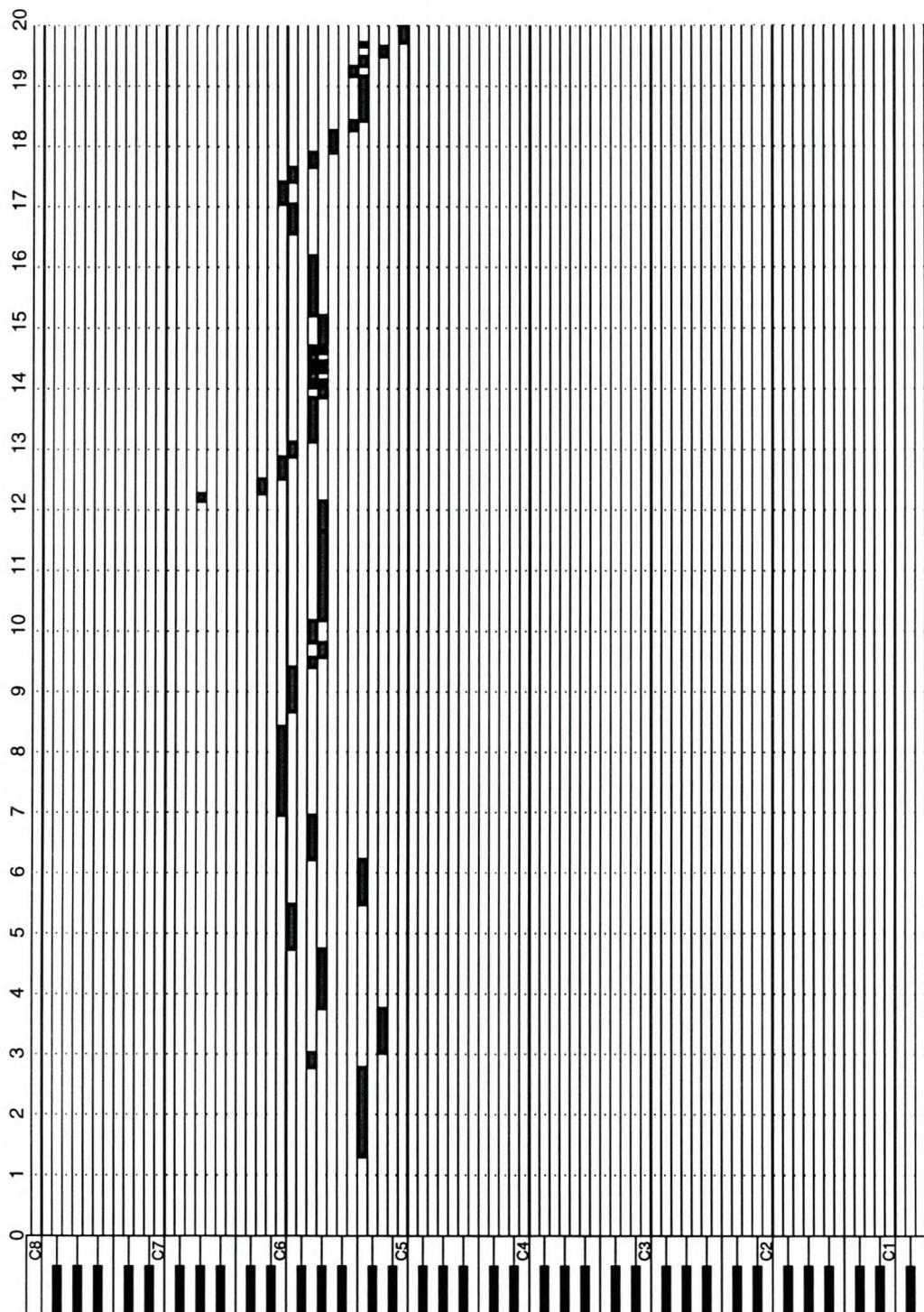


Figure G.2: *Piano roll plot of results for the recorder sample*

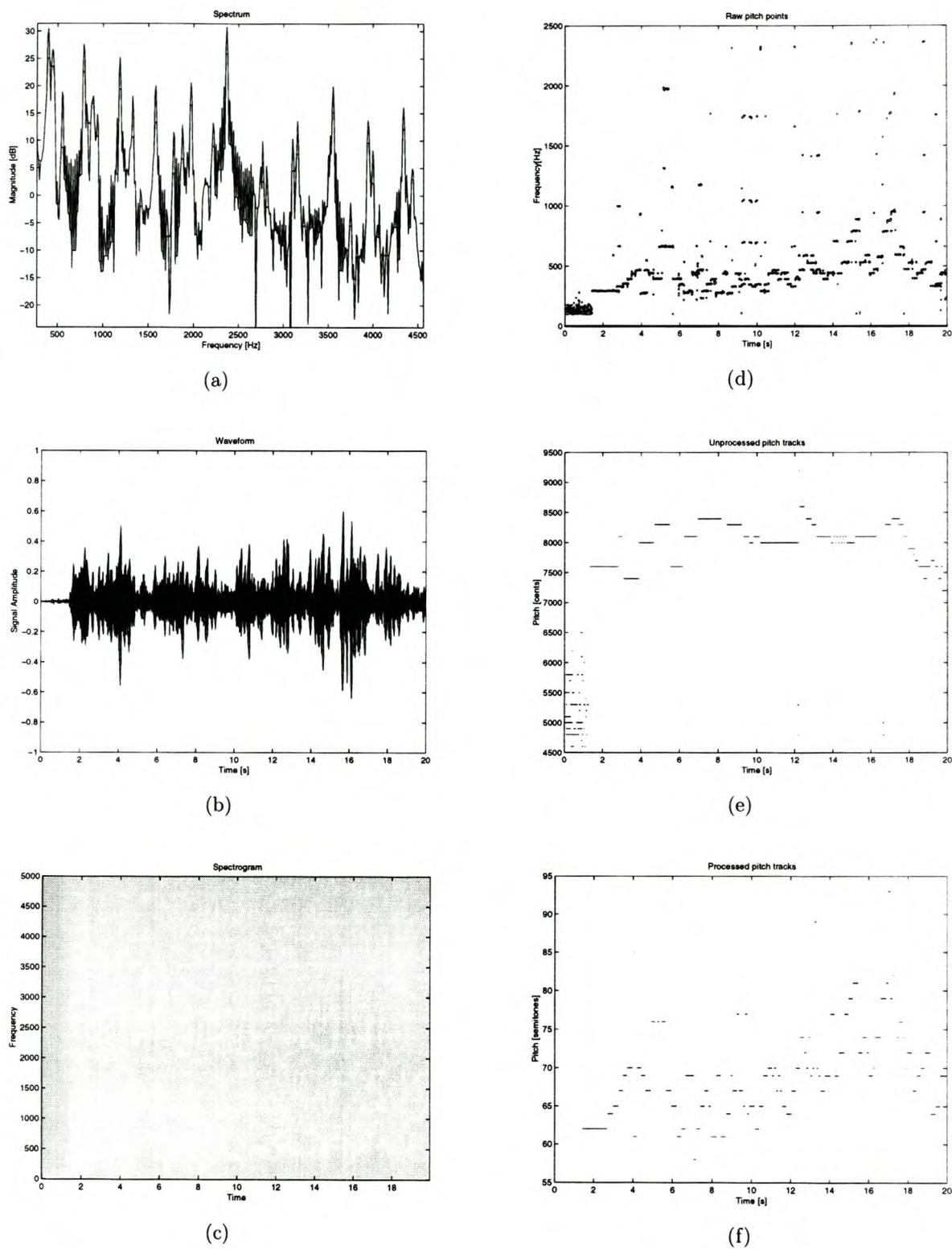


Figure G.3: Steps during the transcription of a violin sample

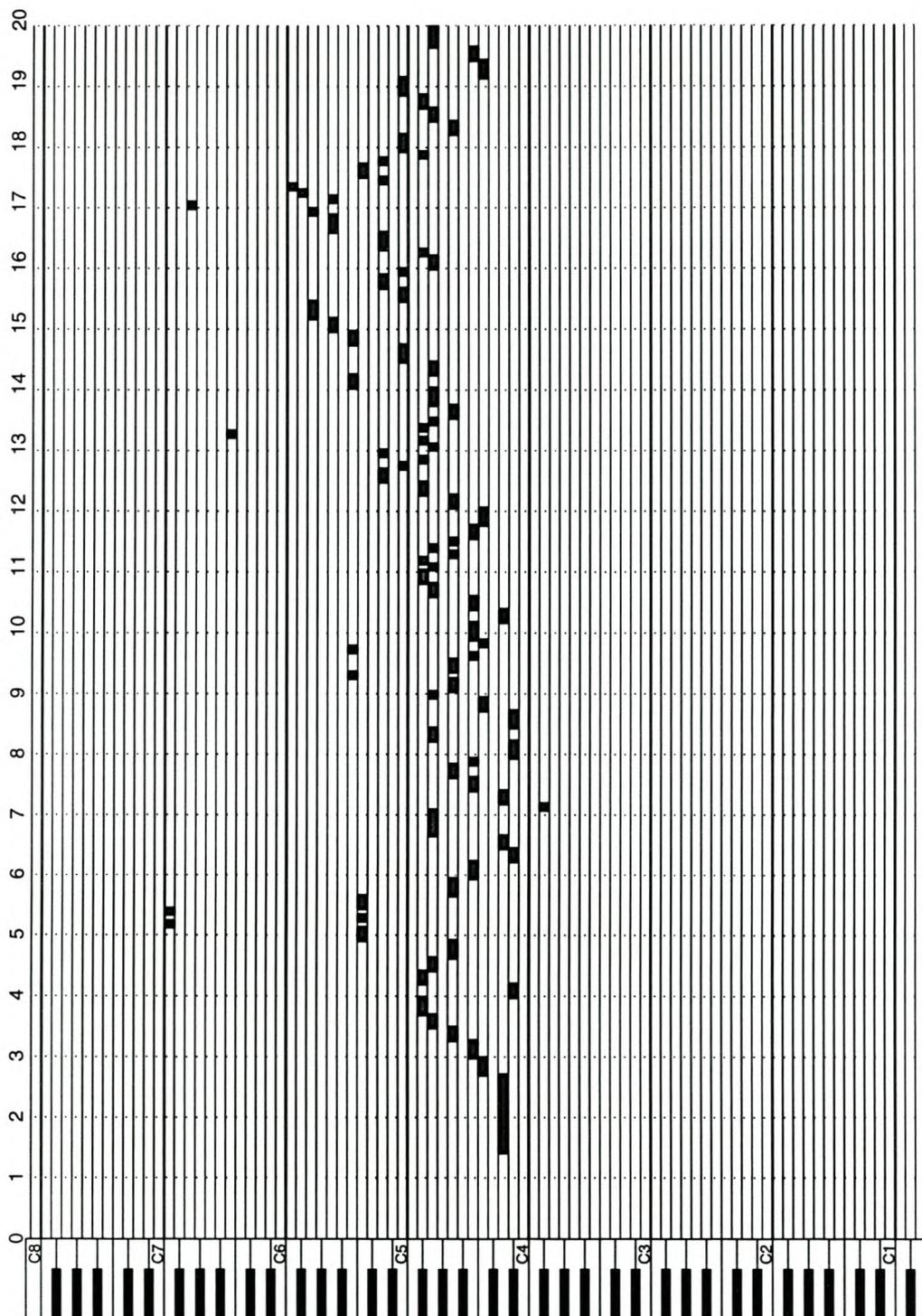
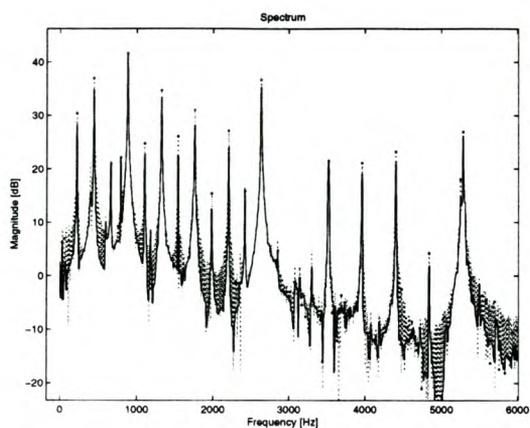
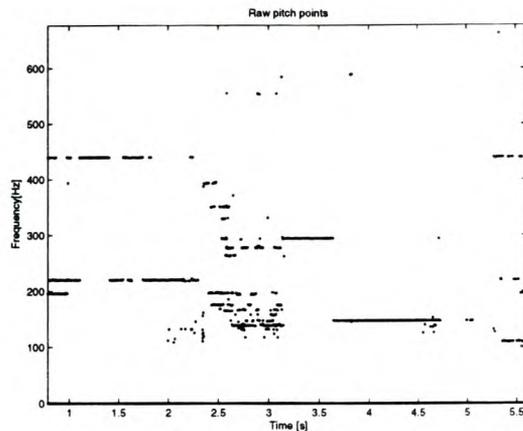


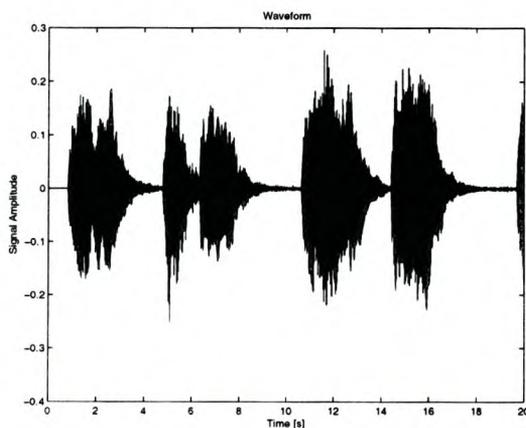
Figure G.4: *Piano roll plot of results for the violin sample*



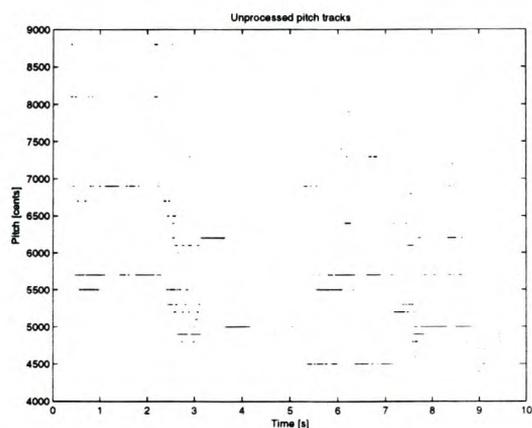
(a)



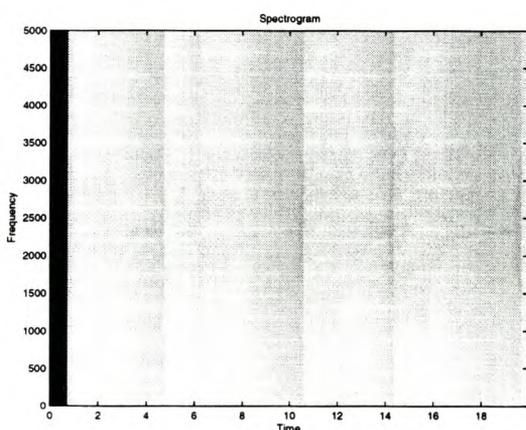
(d)



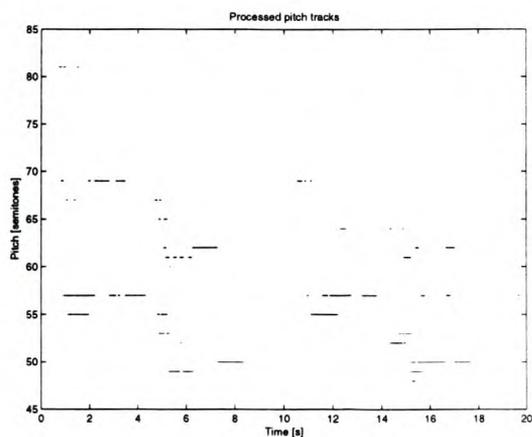
(b)



(e)



(c)



(f)

Figure G.5: Steps during the transcription of a polyphonic organ sample

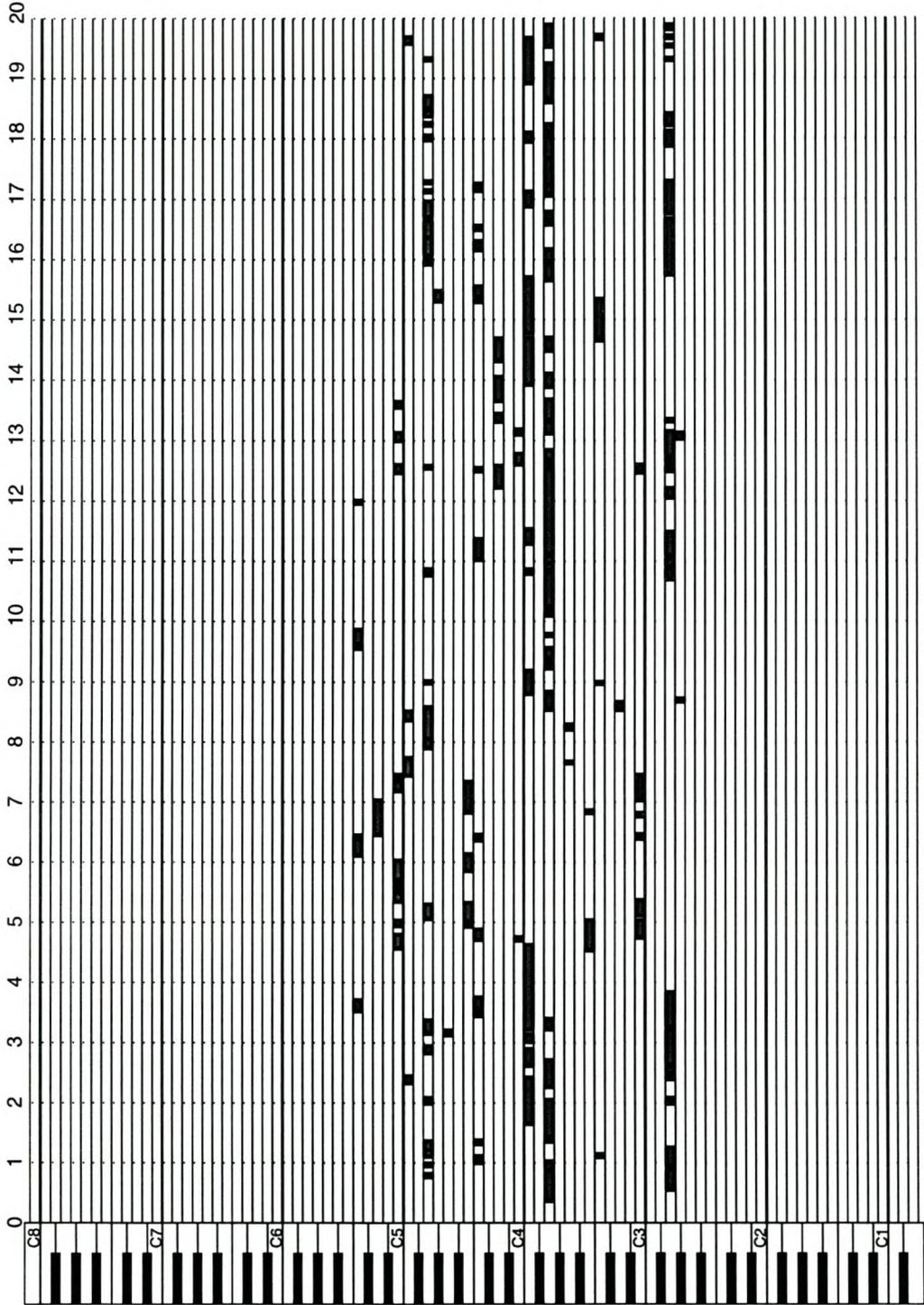


Figure G.6: Piano roll plot of results for the organ sample

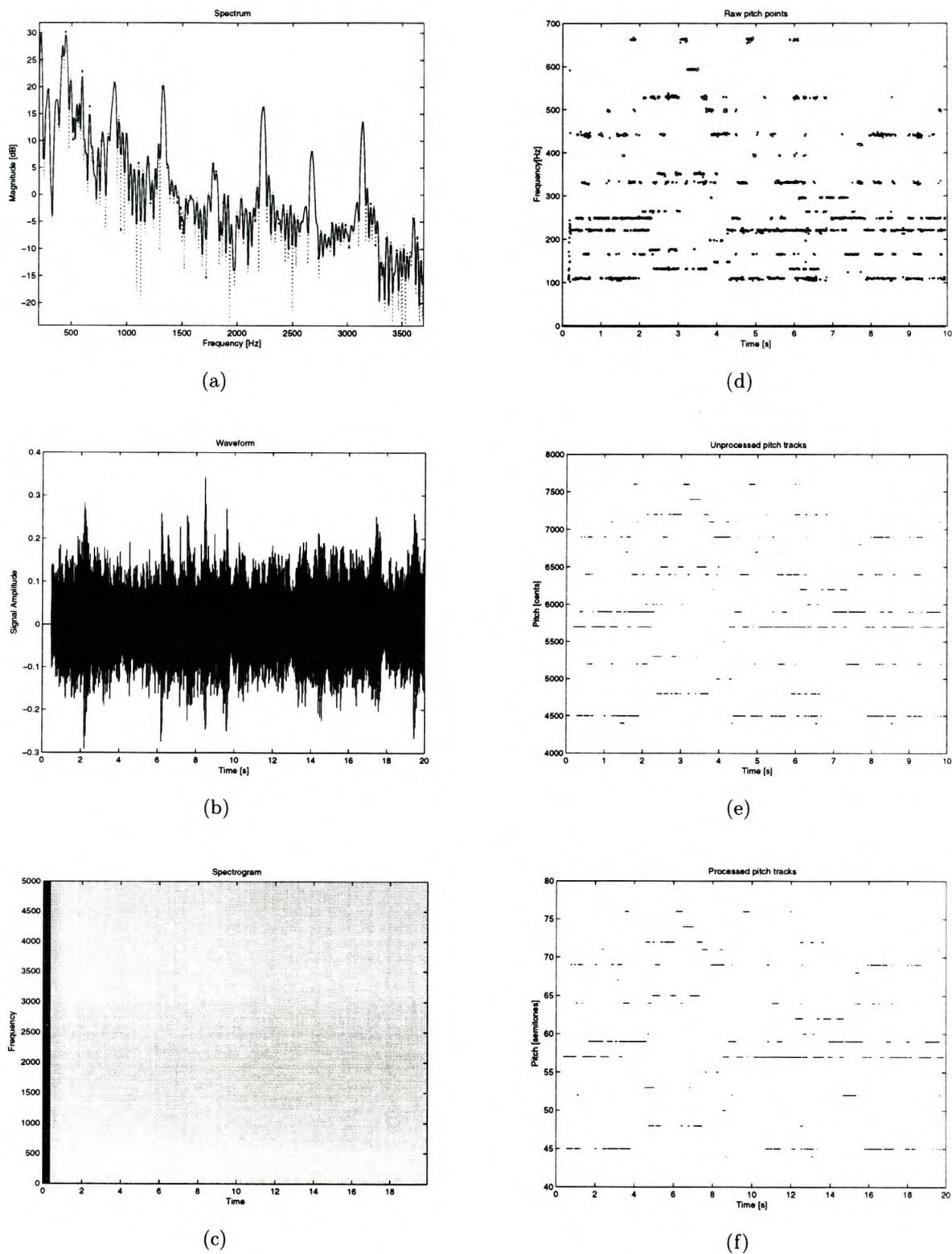


Figure G.7: Steps during the transcription of a polyphonic piano sample

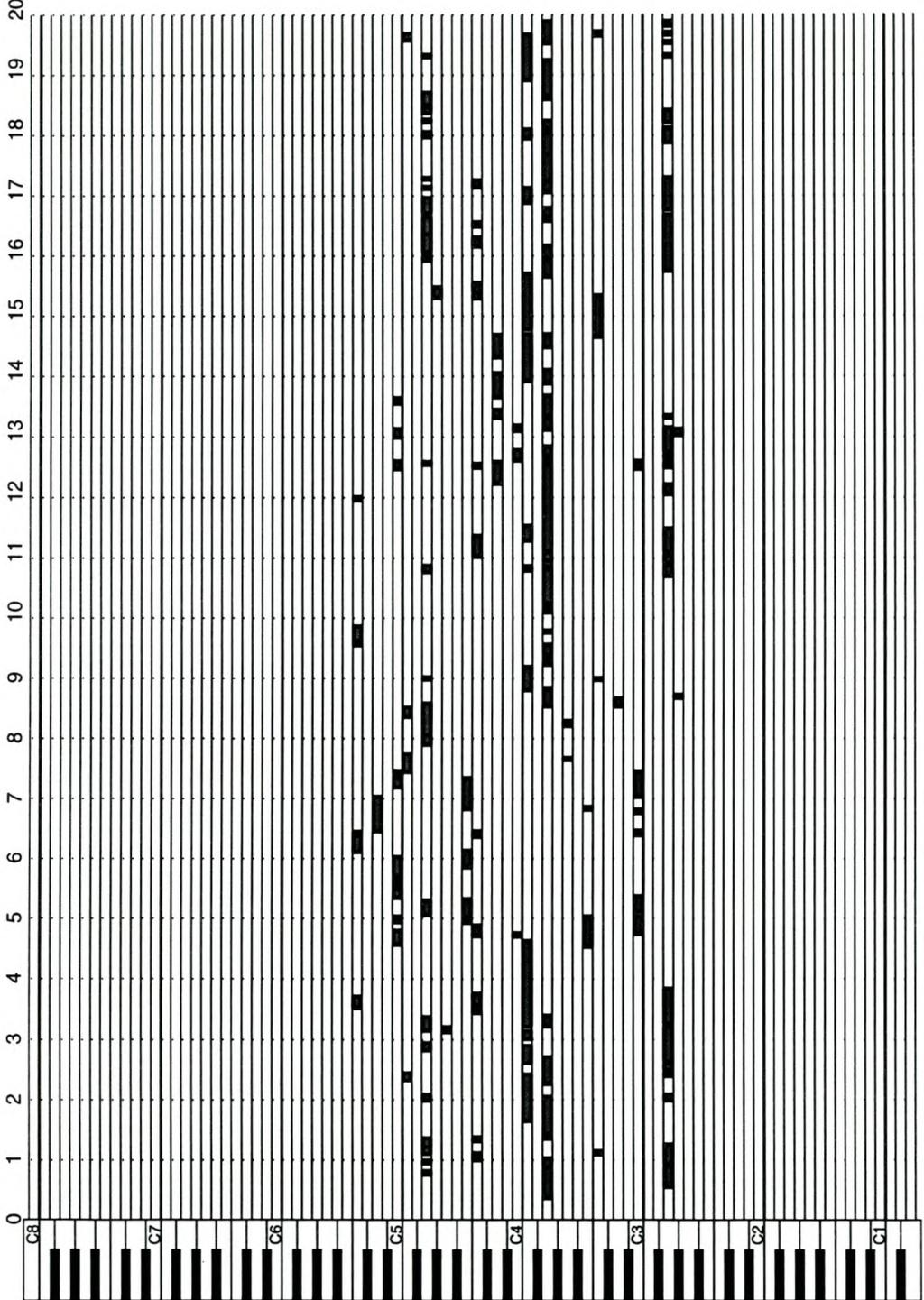


Figure G.8: *Piano roll plot of results for the piano sample*