Universidade de Coimbra
Faculdade de Ciências e Tecnologia
Departamento de Engenharia Informática

Melody Detection in Polyphonic Audio

Rui Pedro Pinto de Carvalho e Paiva

September 2006

Thesis submitted to the
University of Coimbra
in partial fulfillment of the requirements for the degree of
Doctor of Philosophy in Informatics Engineering

This work was carried out under the supervision of

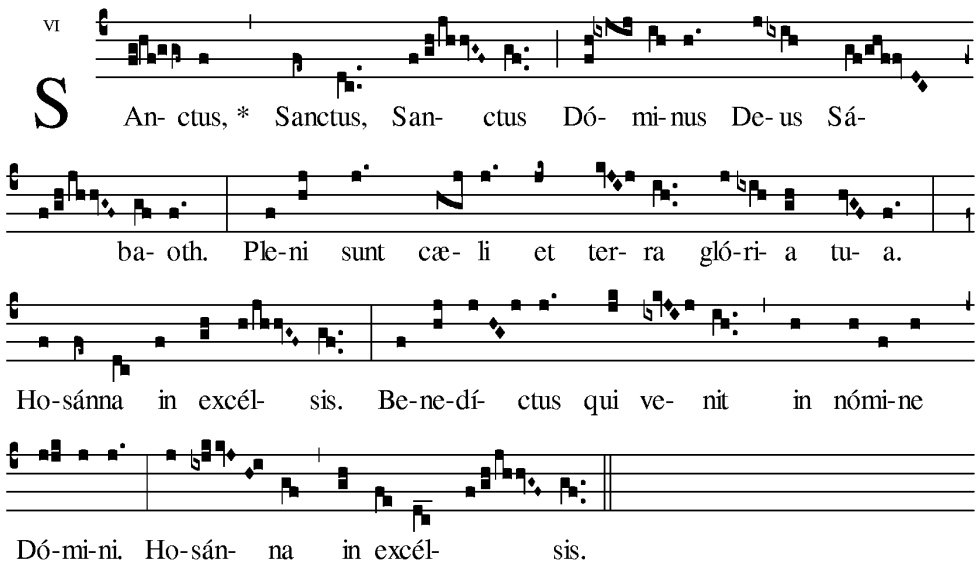
Professora Doutora Maria Teresa Soares Mendes

Professora Catedrática do
Departamento de Engenharia Informática da
Faculdade de Ciências e Tecnologia da
Universidade de Coimbra

and

Prof. Doutor Fernando Amílcar Bandeira Cardoso

Professor Associado do
Departamento de Engenharia Informática da
Faculdade de Ciências e Tecnologia da
Universidade de Coimbra


VI
S
An- ctus, * Sanctus, San- ctus Dó- mi-nus De-us Sá-
ba- oth. Ple-ni sunt cæ- li et ter- ra gló-ri- a tu- a.
Ho-sánna in excél- sis. Be-ne-dí- ctus qui ve- nit in nómi-ne
Dó-mi-ni. Ho-sán- na in excél- sis.

ABSTRACT

In this research work, we address the problem of melody detection in polyphonic audio. Our system comprises three main modules, where a number of rule-based procedures are proposed to attain the specific goals of each unit: i) *pitch detection*; ii) *determination of musical notes* (with precise temporal boundaries and pitches); and iii) *identification of melodic notes*. We follow a multi-stage approach, inspired on principles from perceptual theory and musical practice. Physiological models and perceptual cues of sound organization are incorporated into our method, mimicking the behavior of the human auditory system to some extent. Moreover, musicological principles are applied, in order to support the identification of the musical notes that convey the main melodic line.

Our algorithm starts with an *auditory-model-based pitch detector*, where multiple pitches are extracted in each analysis frame. These correspond to a few of the most intense fundamental frequencies, since one of our basis assumptions is that the main melody is usually salient in musical ensembles.

Unlike most other melody extraction approaches, we aim to explicitly distinguish individual musical notes, characterized by specific temporal boundaries and MIDI note numbers. In addition, we store their exact frequency sequences and intensity-related values, which might be necessary for the study of performance dynamics, timbre, etc. We start this task with the *construction of pitch trajectories* that are formed by connecting pitch candidates with similar frequency values in consecutive frames. The objective is to find regions of stable pitches, which indicate the presence of musical notes.

Since the created tracks may contain more than one note, temporal segmentation must be carried out. This is accomplished in two steps, making use of the pitch and intensity contours of each track, i.e., *frequency* and *salience-based segmentation*. In frequency-based track segmentation, the goal is to separate all notes of different pitches that are included in the same trajectory, coping with glissando, legato and vibrato and other sorts of frequency modulation. As for salience-based segmentation, the objective is to separate consecutive notes at the same pitch, which may have been incorrectly interpreted as forming one single note.

Regarding the identification of the notes bearing the melody, we found our strategy on two core assumptions that we designate as the *salience principle* and the *melodic smoothness principle*. By the salience principle, we assume that the melodic notes have, in general, a higher intensity in the mixture (although this is not always the case). As for the melodic smoothness principle, we exploit the fact that melodic intervals tend normally to

be small. Finally, we aim to *eliminate false positives*, i.e., erroneous notes present in the obtained melody. This is carried out by removing the notes that correspond to abrupt salience or duration reductions and by implementing note clustering to further discriminate the melody from the accompaniment.

Experimental results were conducted, showing that our method performs satisfactorily under the specified assumptions. However, additional difficulties are encountered in song excerpts where the intensity of the melody in comparison to the surrounding accompaniment is not so favorable.

To conclude, despite its broad range of applicability, most of the research problems involved in melody detection are complex and still open. Most likely, sufficiently robust, general, accurate and efficient algorithms will only become available after several years of intensive research.

Keywords: melody detection in polyphonic audio, music information retrieval, melody perception, musicology, pitch detection, conversion of pitch sequences into musical notes, pitch tracking and temporal segmentation, onset detection, identification of melodic notes, melody smoothing, note clustering.

Acknowledgments

"Gratitude is not only the greatest of virtues, but the parent of all the others."

Cicero (106 BC - 43 BC)

Music has been present in my life ever since I was a kid. Although in the beginning this relationship had more to do with hate than with love, I learned to appreciate it, despite those Sunday afternoons when, to my despair, soccer had to stop because mom was calling me to go to choir rehearsal with my sister. But Maestro José Firmino operated a genuine miracle and managed to get that "untuned" nine-year-old kid actually singing and gaining a taste for music.

Thus, combining my educational background on informatics engineering with my interest in music has been a pleasure to me. Therefore, my first words go to my supervisors, Professor Teresa Mendes and Professor Amílcar Cardoso, for the opportunity to perform research in this field and for their support, motivation and friendship.

Thanks go also to CISUC, Center for Informatics and Systems of the University of Coimbra, where most of this work was developed, for the logistic and computer facilities and for the financial support regarding acquisition of material and participation in several scientific conferences in the area. I would like to express my gratitude to all the staff, researchers and collaborators for the pleasant environment and conditions offered. I would like to thank especially my friend Professor Paulo de Carvalho for the fruitful discussions and continuous interest in my work. In the same way, my appreciation goes to Professors António José Mendes and Francisco Câmara Pereira.

Part of this project was carried out at IPEM, Institute for Psychoacoustics and Electronic Music, at the University of Ghent, Belgium. The two months I passed in that multi-disciplinary research group, surrounded with motivated and motivating people, were particularly inspiring. I would like to thank Professor Marc Leman for his openness in accepting me when I was still walking my first steps in this field, as well as for his kindness and incentive. I would also like to especially thank Liesbeth De Voogdt for her support concerning accommodation and settling down in Ghent, Koen Tanghe, Gaëtan Maartens and Dirk Van Steelant for the prolific audio processing discussions and Olmo Cornelis for his friendship and for his guidance and assistance in musicological matters. My gratitude goes, once again, to my supervisors who encouraged me to visit this labora-

tory and established the necessary contacts with Professor Marc Leman.

A work of this kind requires motivation, emotional balance, a lot of effort and some renounces. Hence, I thank my parents for their endless love and confidence, for all their sacrifices in order that I could have a good education in every sense, for teaching me so many things we don't learn at school and for the meals I took at their place when I had no time to cook. This is extended to my dearest sister who has always been there for me, no matter if their babies didn't let her sleep during the night. I also thank my brother-in-law and my little nephews for all the fun we've had together. "Sou Benfica" and a towel is all you need to let the party start!

I am grateful to my friends as well for the good times we have spent together, for the sports, for the "nguenda", for the music and for their patience in so many moments when I could not be with them because I was writing papers or something.

I could not finish without expressing my warmest gratitude to the teachers I've had. I believe that teaching is a most noble and beautiful profession and I was lucky to have had very good masters throughout my life, which loved their job and their students, showing it daily in their diligence and interest. Besides, it is unquestionable that firm and secure first steps pave the way to our future. For this reason, for her love and for the things that no one else taught me and are still strongly rooted in my mind and in my heart, I would like to dedicate a special word to one of my primary school teachers, Miss Maria Amélia Reis Abreu.

Finally, and above all, I send all my love and gratitude to my God, in Whom all my life is founded. You have given me all that I have and all that I am. Borrowing the words of George Herbert, "Thou hast given me so much... Give me one thing more, a grateful heart."

Coimbra, July 31, 2006

Rui Pedro Pinto de Carvalho e Paiva

The research work presented in this thesis was partially supported by the Portuguese Ministry of Science and Technology, under the program PRAXIS XXI.

CONTENTS

Abstract ..... **vii**
Acknowledgments ix
Contents xi
List of Figures xv
List of Tables xvii
List of Algorithms xix
Main Abbreviations xxi
Chapter 1 Introduction **1**
1.1. Motivation and Scope 3
1.1.1. Query-By-Example 5
1.1.2. Query-By-Melody and Melody Detection in Polyphonic Recordings 5
1.1.3. Other Applications of Melody Detection 8
1.2. Objectives and Approaches 9
1.3. Main Contributions 11
1.3.1. Pitch Detection 11
1.3.2. From Pitches to Notes 12
1.3.3. Identification of Melodic Notes 13
1.3.4. List of Publications 13
1.4. Outline of the Dissertation 15
Chapter 2 Melody Detection: Context and Overview **19**
2.1. Music Information Retrieval (MIR) 20
2.1.1. MIR applications 21
2.1.2. MIR Representations and Research Areas 22
2.1.3. MIR Methodological Needs 24
2.1.4. MIR Evaluation Methodologies 25
2.2. Content Analysis and Music Listening 26
2.2.1. Music Content Analysis Paradigms 27

2.2.2. Music and Melody Perception 28
2.3. Melody Definition 37
2.4. Melody Detection in MIR Research 40
2.4.1. Automatic Music Transcription 40
2.4.2. Overview of Research on Melody Detection 44
2.5. Overview of the Proposed Melody Detection System 48
2.6. Test Collections and Evaluation Procedures 51
2.6.1. Acquisition of Ground Truth Data 52
2.6.2. Evaluation Metrics 55
Chapter 3 Pitch Detection 61
3.1. Introduction 63
3.1.1. Harmonic Sounds, Fundamental Frequency and Pitch 63
3.1.2. The Pitch Detection Process 65
3.1.3. Monophonic Pitch Detection 66
3.1.4. Polyphonic Pitch Detection 73
3.2. Pre-Processing: RASTA Processing 79
3.3. Extraction: Auditory-Model-based Pitch Detector 83
3.3.1. Ear Model 85
3.3.2. Channel Periodicity Analysis 93
3.3.3. Periodicity Summarization 95
3.3.4. Salient Peak Detection 96
3.3.5. Illustration of the Algorithm 97
3.4. Post-Processing: SACF Enhancement 97
3.5. Putting It All Together 99
3.6. Experimental Results, Analysis and Conclusions 101
Chapter 4 From Pitches to Notes 111
4.1. Introduction 113
4.1.1. The Note as a Basic Representational Symbol 113
4.1.2. Current Approaches for Note Determination 114
4.2. Pitch Trajectory Construction (PTC) 119
4.2.1. MIDI Quantization 119
4.2.2. Peak Continuation based on Frequency Proximity 120

4.2.3. Track Inactivity 121
4.2.4. Tackling Ambiguities 122
4.2.5. Elimination of Short Tracks 123
4.2.6. Reassignment of Unused Pitch Candidates 123
4.2.7. Putting It All Together 123
4.3. Frequency-Based Track Segmentation 126
4.3.1. Note Segmentation 126
4.3.2. Note Labeling 132
4.3.3. Merging of Simultaneous PCFs with Equal MIDI Note Numbers 134
4.3.4. Putting It All Together 134
4.4. Salience-Based Track Segmentation 136
4.4.1. Candidate Segmentation Points 136
4.4.2. Onset Detection 138
4.4.3. Validation of Candidate Segmentation Points 142
4.4.4. Segmentation after Melody Identification 143
4.4.5. Putting It All Together 144
4.5. Putting It All Together 146
4.6. Experimental Results, Analysis and Conclusions 147
Chapter 5 Identification of Melodic Notes 155
5.1. Introduction 157
5.1.1. Approaches based on Full Source Separation 157
5.1.2. Approaches based on Figure-Ground Organization 158
5.2. Elimination of Ghost Harmonically-Related Notes 159
5.2.1. Exploiting Harmonicity 159
5.2.2. Exploiting Common Fate 160
5.2.3. Integration of Harmonicity and Common Fate 161
5.2.4. Putting It All Together 162
5.3. Selection of the Most Salient Notes 163
5.3.1. Elimination of Non-Dominant Notes 164
5.3.2. Resolution of Note Overlaps 165
5.3.3. Putting It All Together 167
5.4. Melody Smoothing 169

5.4.1. Octave Correction 169
5.4.2. Resolution of Abrupt Note Transitions 169
5.4.3. Gap Filling 171
5.4.4. Note Timing Restoration 171
5.4.5. Putting It All Together 172
5.5. Elimination of Spurious Accompaniment Notes 174
5.5.1. Analysis of the Salience Contour 174
5.5.2. Analysis of the Duration Contour 175
5.5.3. Putting It All Together 175
5.6. Note Clustering 177
5.6.1. Acoustical Correlates of Timbre 178
5.6.2. Feature Extraction 179
5.6.3. Feature Selection and Dimensionality Reduction 185
5.6.4. Clustering 186
5.6.5. Putting it All Together 189
5.7. Putting It All Together 190
5.8. Experimental Results, Analysis and Conclusions 192
Chapter 6 Conclusions and Perspectives 203
6.1. Summary and Conclusions 203
6.2. Perspectives for Future Research 205
Bibliography 209
Appendix A Other Evaluated Pitch Detection Approaches 221
A.1. STFT-based Harmonic Analysis 221
A.2. Probabilistic Approach 224
Appendix B Description of Song Excerpts 229

List of Figures

Figure 2.1. Melody detection system overview. 49
Figure 3.1. Time and frequency-domain illustration of a harmonic sound. 64
Figure 3.2. Overview of the monophonic pitch detection process. 66
Figure 3.3. Envelope periodicity: a) and b) original time-domain signal and respective magnitude spectrum; c) and d) half-wave rectified signal and spectrum; e) and f) amplitude envelope and spectrum. 71
Figure 3.4. Spectra of an isolated harmonic sound and of a mixture of sounds..... 74
Figure 3.5. Additive noise suppression by spectral subtraction. 82
Figure 3.6. Filtered temporal signal. 83
Figure 3.7. Auditory-model based pitch detector (AMPD). 85
Figure 3.8. Frequency sensitiveness along the basilar membrane. 87
Figure 3.9. Lyon's ear model. 88
Figure 3.10. Frequency response of cochlear filters. 91
Figure 3.11. Automatic gain control. 92
Figure 3.12. Output after four stages of AGC. 92
Figure 3.13. Cochleagram of a 2.5s' saxophone riff. 93
Figure 3.14. Results of the four stages of the AMPD algorithm. 97
Figure 3.15. SACF Enhancement. 99
Figure 3.16. Response of the AMPD to fast pitch variations. 106
Figure 4.1. Look-ahead procedure. 122
Figure 4.2. PTC algorithm. 124
Figure 4.3. Results of the PTC algorithm. 126
Figure 4.4. Oscillation filtering. 129
Figure 4.5. Filtering of delimited sequences. 129
Figure 4.6. Glissando filtering. 130
Figure 4.7. Final short note filtering. 131
Figure 4.8. Results of the frequency-based track segmentation algorithm. 135

Figure 4.9. Results of salience-based track segmentation: initial candidate points. 138
Figure 4.10. Results of onset detection. 141
Figure 4.11. Comparison of onset results using the RDF and FOD functions. 141
Figure 4.12. Results of the salience-based track segmentation algorithm 143
Figure 4.13. Results of the salience-based track segmentation algorithm: acceptance of clear salience minima. 144
Figure 5.1. Similarity analysis of frequency curves. 161
Figure 5.2. Similarity analysis of salience trends. 161
Figure 5.3. Results of the elimination of ghost notes 163
Figure 5.4. Definition of segments in a song excerpt. 164
Figure 5.5. Types of note overlapping 166
Figure 5.6. Results of the algorithm for selection of the most salient notes. 167
Figure 5.7. Regions of smoothness. 170
Figure 5.8. Results of the melody-smoothing algorithm. 172
Figure 5.9. Pitch salience contour (jazz3 excerpt). 174
Figure 5.10. Results of the algorithm for elimination of spurious notes. 175
Figure 5.11. Detection of harmonics from the correlogram. 180
Figure 5.12. Results of the note clustering algorithm (jazz3 excerpt). 189
Figure A.1. Results of the STFT-based harmonic analysis. 224
Figure A.2. Frequency response of the melodic band-pass filter. 225
Figure A.3. Results of the probabilistic pitch detector. 228

List of Tables

Table 2.1. Description of used song excerpts. Excerpts 1-11: personal database; excerpts 12-12: MIREX'2004 training set. 54

Table 3.1. AMPD parameters. 101

Table 3.2. Comparison of pitch detection algorithms. Algorithms are sorted by raw pitch detection accuracy. 102

Table 3.3. AMPD results: pre and post-processing and single-pitch detection. 103

Table 4.1. PTC parameters. 126

Table 4.2. Parameters for frequency-based track segmentation. 135

Table 4.3. Parameters for salience-based track segmentation (line 1: detection of candidate segmentation points; lines 2 to 11: onset detection; line 13: validation of candidate points; line 14: segmentation after melody identification. 145

Table 4.4. Note determination results: accuracy for PTC and trajectory segmentation, with and without tuning compensation. 148

Table 4.5. Results for frequency-based track segmentation. 149

Table 4.6. Results for salience-based track segmentation. 151

Table 5.1. Parameters for the elimination of ghost harmonically-related notes. 162

Table 5.2. Parameters for extraction of salient notes. 168

Table 5.3. Melody smoothing parameters. 173

Table 5.4. Parameters for elimination of spurious notes. 177

Table 5.5. Note clustering parameters. 191

Table 5.6. Results for the elimination of ghost harmonically-related notes. 193

Table 5.7. Results of melody detection: selection of salient notes and melody smoothing. 194

Table 5.8. Results of the melody detection system: elimination of accompaniment notes. 196

Table 5.9. Results of the MIREX'2004 evaluation. 199

Table 5.10. Results of the MIREX'2005 evaluation. 201

List of Algorithms

Algorithm 3.1. Pitch detection. 100
Algorithm 4.1. Pitch trajectory construction. 124
Algorithm 4.2. Frequency-based track segmentation. 134
Algorithm 4.3. Salience-based track segmentation. 144
Algorithm 4.4. From pitches to notes. 146
Algorithm 5.1. Elimination of harmonically-related notes. 162
Algorithm 5.2. Selection of the most salient notes. 167
Algorithm 5.3. Melody extraction using melodic smoothness. 172
Algorithm 5.4. Elimination of spurious notes. 176
Algorithm 5.5. Note clustering. 189
Algorithm 5.6. Identification of melodic notes. 191

Main Abbreviations

ACF	AutoCorrelation Function	(defined on page 47)
AMPD	Auditory-Model-based Pitch Detector	(page 11)
BPF	Band-Pass Filter	(page 48)
DFT	Discrete Fourier Transform	(page 221)
EMD	Electronic Music Distribution	(page 2)
ETF	Equal Temperament Frequency	(page 46)
F0	Fundamental Frequency	(page 6)
FFT	Fast Fourier Transform	(page 68)
GMM	Gaussian Mixture Model	(page 45)
HMM	Hidden-Markov Model	(page 48)
ISMIR	International Conference (Symposium) on Music Information Retrieval	(page 15)
M04	MIREX'2004 database	(page 53)
MCA	Music Content Analysis	(page 22)
MCNA	Melodic Chroma Note Accuracy	(page 58)
MCPA	Melodic Chroma Pitch Accuracy	(page 57)
MIDI	Musical Instrument Digital Interface	(page 7)
MIR	Music Information Retrieval	(page 4)
MIREX	Music Information Retrieval Evaluation eXchange	(page 15)
MRNA	Melodic Raw Note Accuracy	(page 58)
MRPA	Melodic Raw Pitch Accuracy	(page 57)
ORNA	Overall Raw Note Accuracy	(page 58)
ORPA	Overall Raw Pitch Accuracy	(page 56)
PCA	Principal Component Analysis	(page 177)
PCF	Piecewise-Constant Function	(page 126)
PDB	Personal DataBase	(page 53)
PDF	Probability Density Function	(page 45)

PTC	Pitch Trajectory Construction	(page 50)
QBE	Query-By-Example	(page 5)
QBH	Query-By-Humming	(page 6)
QBM	Query-By-Melody	(page 5)
RASTA	RelAtive SpecTrAl	(page 17)
SACF	Summary AutoCorrelation Function	(page 62)
SNR	Signal-to-Noise Ratio	(page 12)
STFT	Short-Time Fourier Transform	(page 14)

Chapter 1

INTRODUCTION

"There is sweet music here that softer falls
Than petals from blown roses on the grass,
Or night-dews on still waters between walls
Of shadowy granite, in a gleaming pass;
Music that gentlier on the spirit lies,
Than tired eyelids upon tired eyes;
Music that brings sweet sleep down from the blissful skies."

Alfred Lord Tennyson, "The Song of the Lotos-Eaters", 1832

Wherever man is, there is music. Music was, is and will always be present in the lives of people, both individually and socially, through the cultural, leisure, religious or professional dimensions of existence. As a means of celebration, music has always accompanied man's festive moments. In the expression of human religiosity, music is synonymous of prayer and draws man closer to the transcendent and to the infinite. In sport activities, music can be used to keep an athlete motivated, relax him or help him to focus. Music can be approached as an aid in the therapy of nervous disturbances or even in the improvement of student performances.

Music is a most eloquent form of communication, expressing "that which cannot be put into words and that which cannot remain silent" (Victor Hugo). Composers are the masters of such powerful language, acting as our interlocutors and using it to portray what we and them feel incapable of communicating solely by words: love, passion, tenderness, anguish and serenity, joy and sorrow; in one word, the four seasons of the soul. Their language then becomes our language: by listening to music, emotions and memories, laughter and tears, thoughts and reactions, are awakened. We associate music with the most unique moments of our lives and music is part of our individual and social imaginary. Yes, "life has a soundtrack" [Gomes, 2005]... And, for that matter, mankind also has, as "the history of a people is found in its songs" (George Jellinek).

Music is all around us. It is in the street and in the bus, in the elevator, in the gym, on radio and TV, in church, in supermarkets, in pubs and, strikingly, on the Internet.

Given its major importance in all human societies throughout history and particularly nowadays, music plays a role in the world economy. In fact, the music industry runs, only in the USA, an amount of money in the order of several billion US dollars per year. For illustration, it is estimated that Apple iTunes[1] sells approximately 1.25 million songs everyday. Since the service was launched (April 2003), until the beginning of 2005, around 250 million songs had been sold in total [TechWhack, 2005]. At 99 USD cents per song, this figure amounts to $1,237,500 per day and $451,687,500 per year.

These days, digital music is available in many and different forms, places and contexts. Indeed, as a consequence of recent technological innovations, there has been a tremendous growth in the Electronic Music Distribution (EMD) industry. Factors like the widespread access to the Internet, bandwidth increasing in domestic and mobile accesses, the development of compact audio formats with CD or near CD quality (e.g., mp3, wma), portable music devices, peer-to-peer networks (e.g., Napster[2], Kazaa[3]), online music stores (e.g., iTunes, OD2[4]) or music identification platforms (e.g., Shazam[5], 411-Song[6]) have given a great contribution to that boom. Presently, it is expected that the number of digital music archives, as well as their dimension, grow significantly in the near future, both in terms of music database size and in number of genres covered. This situation poses new perspectives and challenges to music librarians and service providers.

In this introductory chapter, we present the main motivations, objectives and contributions of this research work, and the overall organization of the dissertation. The chapter is structured as described in the following paragraphs.

Section 1.1. Motivation and Scope

First of all, we introduce the main motivations and scope of this project. The problem of music retrieval is presented and some of its modalities are described, namely query-by-example and query-by-melody. The relevance of melody detection in polyphonic audio to several application domains is then discussed.

---

[1] http://www.apple.com/itunes/

[2] http://www.napster.com/

[3] http://www.kazaa.com/

[4] Acronym for On Demand Distribution: http://www.od2.com/

[5] http://www.shazam.com/

[6] http://www.411song.com/

Section 1.2. Objectives and Approaches

In the second section, we describe our main objectives and briefly sketch the overall methodology.

Section 1.3. Main Contributions

The main contributions of this work are then summarized in connection with the three main modules of our system: pitch detection, conversion of pitch sequences into musical notes and identification of melodic notes. The publications that resulted from this project are listed and briefly described.

Section 1.4. Outline of the Dissertation

We end this chapter with the roadmap of the dissertation. The structure of the document is presented and the content of each chapter is outlined.

1.1. Motivation and Scope

Peter went to a bookstore. His girlfriend's birthday is approaching and a few days ago she said something about historical romances. He goes to the corresponding section of the shop and picks a few books that seem interesting. Ambient music sounds softly and, while reading the synopses, he unconsciously starts to hum the listened melodies. At some point, an unknown song catches his attention. "Nice sound", he thinks. After a few repetitions of the chorus, he has more or less memorized it and starts to accompany it quietly. He continues checking the synopses and then decides for some particular book. After leaving the shop, the unknown song is still sounding in his mind. "I wonder what song this is; sounds like Scottish folk or something". To satisfy his curiosity, he uses his mobile phone to call a music identification service and sings the parts of the melody he remembers. The sung excerpt is submitted to the music search engine of the service provider and a few seconds later an html-type message is returned, with links to brief sound summaries of the spotted songs. He then starts his mobile phone's Internet browser, follows the links, listens to the first few excerpts and finds out that the piece he looks for belongs to the last CD of "The Battlefield Band". He notices he is close to a CD store. He goes in and buys the CD.

The above scenario illustrates the fact that any large music database, or, generically speaking, any multimedia database, is only truly useful if users can find what they are looking for in an efficient manner. Furthermore, it is important as well that the organization of such databases be performed as objectively and efficiently as possible. Current ever-expanding music repositories contain tremendous amounts of songs. For instance,

by 2000, "a typical database of titles (e.g., Sony Music[7]) contain[ed] about 500000 titles (...). A database containing all tonal music recordings would probably reach 4 millions titles. Adding ethnic music and non-western types of music would probably double or triple this number. Every month, about 4000 CDs [were] created in Western countries" [Pachet and Cazaly, 2000]. The same quoted authors also report that a music database such as Amazon's[8] was, by that time, organized into a taxonomy of 719 genres. These days, online music stores such as iTunes have repositories with over 2 million songs[9].

Presently, whether it is the case of digital music libraries, the Internet or any music database, search and retrieval is carried out mostly in a textual manner, based on categories such as artist, title or genre. In spite of its unquestionable usefulness and wide acceptance, this strategy leads to a certain number of difficulties, both for service providers, in what concerns the manual assignment of such tags, and customers, in terms of database search in transparent and intuitive ways, in accordance with users' preferences. In effect, "music's preeminent functions are social and psychological", and so "the most useful retrieval indexes are those that facilitate searching in conformity with such social and psychological functions. Typically, such indexes will focus on stylistic, mood, and similarity information" [Huron, 2000].

Therefore, in order to overcome the described limitations, research is being conducted in an emergent and promising field named Music Information Retrieval (MIR). MIR is a strongly inter-disciplinary research area that has evolved from the necessity to manage huge collections of digital music for "preservation, access, research and other uses" [Futrelle and Downie, 2003]. This is indubitably a field with tremendous potential for applications.

Generally speaking, research is progressing on topics such as automatic music classification and feature extraction, audio fingerprinting, music recommendation, automatic music transcription, melody detection, song database indexing, music representations or user interface design, to name but a few.

In particular, content analysis and similarity assessment and retrieval in audio song databases are receiving significant attention (e.g., [Vignoli and Pauws, 2005; Aucouturier and Pachet, 2004; Berenzweig *et al.,* 2003; Tzanetakis, 2002; Pampalk, 2001; Logan and Salomon, 2001; Yang, 2001; Welsh *et al.,* 1999; Bainbridge *et al.,* 1999]).

In some of those systems, as for example the one described in [Pampalk, 2001], graphical user interfaces based on the metaphor of geographic maps are employed to group songs according to their resemblance: islands denote musical genres, which are "geographically" organized in such a way that songs from similar genres are "physically"

---

[7] http://www.sonymusic.com/

[8] http://www.amazon.com/

[9] http://www.apple.com/itunes/overview/

close together.

In others, the goal is to allow the creation of musical queries through examples supplied by the user, for instance, by humming, whistling or singing the melody to search for - a process denominated query-by-melody (QBM) - or by specifying an excerpt analogous in some way to what is being looked for, based on search criteria such as rhythm, genre, instrumentation or tonality - designated as query-by-example (QBE). The former corresponds to the scenario imagined at the beginning of this section. There, melody assumes particular relevance. In fact, despite other implicit tasks (such as, for example, song summarization), melody detection in polyphonic recordings is a basic requirement for query-by-melody, as will be discussed in the following paragraphs.

1.1.1. Query-By-Example

Regarding query-by-example, this search scheme is most useful when users do not know the melody, when melodies simply do not exist (e.g., in types of music such as electro-acoustic music) or when users are more interested in other musical features such as rhythm, tonality or even lyrics[10].

Moreover, this mechanism offers interesting possibilities for the discovery of new music complying with the personal, social or psychological purposes of the search [Vignoli and Pauws, 2005; Celma *et al.*, 2005; Pampalk, 2001; Welsh *et al.*, 1999]. Namely, a music store may recommend new music to its customers based on user preference profiles, a movie director may look for a soundtrack that reflects the emotional context of a scene or an aerobic instructor may be interested in songs with a certain tempo, regardless of melody or genre. This can be a daunting undertaking if we think of the thousands or even millions of songs, organized sometimes in tens or hundreds of different and often non-uniform genres that many music libraries contain.

1.1.2. Query-By-Melody and Melody Detection in Polyphonic Recordings

As for QBM, this is an intuitive way of searching for a musical piece, since melody humming, whistling, "syllabbling" or singing are natural habits of humans. Furthermore, it frequently happens that we want to find a song and the only thing we remember is a small fragment of it, for example, the chorus, rather than the title or the performer.

---

[10] Some music lyric search engines, e.g., *LetsSingIt.com*, are available on the web. However, in such systems the lyrics must be manually annotated (or provided by the authors or recording companies), rather than automatically extracted. This would be the ultimate goal to accomplish in this search strategy. Such task entails several complex research problems under the topic of automatic singing speech recognition.

Thus, several techniques have been proposed, aiming to permit song retrieval via aural melodic queries, most of them hummed, i.e., query-by-humming (QBH)[11], but also sung, i.e., query-by-singing (QBS) [Parker, 2005; Batke *et al.,* 2004; Shih *et al.,* 2003; Song *et al.,* 2002; Chai, 2001; Birmingham *et al.,* 2001; Bainbridge *et al.,* 1999; Rolland *et al.,* 1999; Kornstadt, 1998; McNab *et al.,* 1996b; Ghias *et al.,* 1995; Kageyama *et al.,* 1993].

In effect, due to the technical idiosyncrasies of query processing, hummed queries, rather than whistled or sung ones, are preferred in most of the work published so far. This comes from the fact that sung signals are generally more difficult to analyze than hummed signals (namely regarding pitch[12] detection[13] accuracy in the processing of the singing voice or the treatment of octave errors, as will be discussed in Chapter 3). Hence, when we talk about QBM, most of the time we are actually referring to QBH. Anyway, we prefer the term QBM for its generality. Alternatively, systems for QBS require sung queries to consist of discrete notes separated by silence or to be created using particular syllables such as 'ta' or 'da' that are easy to segment into individual notes, as pointed out in [Kim *et al.,* 2000].

In the implementation of robust and efficient QBM mechanisms, factors like query construction, melody extraction, melody representation and melody matching are crucial [Chai, 2001, pp. 3]. For instance, besides pitch detection accuracy, approaches must be robust concerning user's imperfections in the creation of queries, e.g., singing ability, memory flaws, humming "off-key", etc., [Birmingham *et al.,* 2001]. As a consequence, in melody matching, melody similarity evaluation must cope with such distortions and still remain computationally efficient. This is a well-studied information retrieval problem, usually tackled by approximate string matching algorithms, e.g., [Cahill and Ó Maidín, 2005; Hofmann-Engl, 2003; Grachten *et al.,* 2002; Chai, 2001; Lemström and Perttu, 2000; Orpen and Huron, 1992; Wagner and Fischer, 1974]. Distortions are also handled by melody representations such as melodic contours or intervallic representations. These are normally used and allow for singing transpositions or distortions such as raising or lowering the pitch of a few notes [Chai, 2001, pp. 32]. Moreover, the use of melody contours is motivated by the ways humans remember music and, particularly, melodies. Namely, Jay Dowling found out that melodic contours are easier to remember than

---

[11] The Themefinder system [Kornstadt, 1998] only supports text format queries and so, strictly speaking, is not a QBH tool: users have to manually input the text string in conformity with a specified format. The application constitutes a web interface to the Humdrum toolkit, a generic music representation framework, developed by David Huron [Huron, 1997]. Anyhow, the Themefinder is very famous for its organization and features for symbolic data processing.

[12] For language convenience, we will use the term pitch indistinctly of fundamental frequency (F0) throughout this document, though the former is a perceptual variable, whereas the latter is a physical one (see Chapter 3). This "abuse" occurs in most of the related literature and, for the purposes of the present research work, no ambiguities arise from it.

[13] Besides "pitch detection", other terms usually employed in the literature are pitch tracking or pitch estimation (regardless of using probabilistic approaches or not).

exact melodies [Dowling, 1978] (cited in [Chai, 2001, pp. 20]).

In relation to melody extraction, this is a partly solved issue for many of the existing platforms. Indeed, almost all current QBM applications are restricted to the MIDI[14] domain, using files where the melody is generally available in a separate channel, identified by labels such as "melody", "lead" or "vocal". Therefore, the main issues often concern redundancy reduction, namely theme or motive extraction. This is a complex and important topic, which, among other advantages, allows a more efficient matching, given that themes are much smaller than entire pieces [Meek and Birmingham, 2001].

However, when the melody is not explicitly separated in MIDI files, additional difficulties arise. In such cases, the query can be looked for in each of the individual channels, either separate or simultaneously (e.g., [Doraisamy and Rüger, 2001; Dovey, 2001; Lemström and Perttu, 2000; Francu and Nevill-Manning, 2000]). Furthermore, some sort of monophonic reduction may be undertaken when the melody is conveyed in a polyphonic track. Several algorithms have been devised to identify the melody in MIDI files (e.g., [Uitdenbogerd, 2002; Francu and Nevill-Manning, 2000]), exploiting important aspects of melody perception.

Besides MIDI files, QBM can be conducted in monophonic audio as well. But despite the previously mentioned difficulties, monophonic pitch detection is usually considered "practically a solved problem" [Klapuri, 2004, pp. 3]. Hence, melody extraction is not so complicated, at least in comparison to polyphonic performances[15].

Yet, few "real-world" songs are strictly monophonic. Instead, they enclose rich textures, often containing a soloist and harmonic and/or percussive accompaniment. This is the most common kind of musical material and also the one typical users are more interested in: polyphonic and multi-instrumental audio musical pieces, usually obtained from CDs or stored in audio formats such as mp3. Clarifying this point, there are several target audiences for music retrieval tools, each of them with specific requirements and needs: musicologists, composers, music librarians, music shop customers, Internet users, etc., [Uitdenbogerd, 2002]. By *typical users* we mean active listeners, having or not (customarily not) formal musical education, who buy CDs, look for music on the Internet, often want to locate half-remembered songs or to discover new music with specific characteristics or similarities to other work. Consequently, limiting QBM to MIDI files or monophonic recordings places important usability questions.

---

[14] Acronym for Musical Instrument Digital Interface. MIDI files contain linear, time stamped sequences of events. For the purposes of this work, we can say, in simplistic words, that MIDI is a symbolic format for representing music. We are aware of only a few attempts towards QBM in audio databases, so far with incipient results, e.g., [Pikrakis and Theodoridis, 2005; Song *et al.*, 2002].

[15] By *polyphonic music* we refer to musical pieces where several sources are simultaneously present (e.g., vocals, guitar, percussion, etc.), rather than to the more precise theoretical concept, i.e., a type of music in which the individual voices move independently of one another, in contrast to *monophonic music* (one single voice) or *homophonic music* (in which all the voices move more or less together).

Despite the challenges described above for the MIDI domain, we argue that deriving melody representations from polyphonic audio files is a more demanding task since, in the MIDI realm, all the notes, as well as their timings, are already known. This simplifies the extraction of the melody even when it is not directly available. On the other hand, querying polyphonic recordings requires that some sort of melody representation be extracted beforehand, which increases the complexity level of the problem. In effect, polyphonic audio recordings are typically multi-timbral and have many simultaneously sounding notes. Additionally, several instruments (including the singing voice), each one with different and varying spectral properties, interfere significantly with each other, giving rise to complex spectra.

Melody representations for polyphonic audio may be acquired in line with two basic strategies: the explicit extraction of melodic notes (after which melodic or intervallic contours could be obtained) or the development of more abstract and goal-oriented representations. The latter approach is pursued by Jungmin Song *et al.* [Song *et al.*, 2002] in a system for QBH in polyphonic audio databases. Instead of explicitly extracting the melody, they define a mid-level representation consisting of a sequence of audio segments, each containing a set of note candidates. Then, a variation of dynamic programming is employed for matching the query with the melody mid-level representation. Results are, however, incipient.

On the other hand, explicit melody detection, despite being more complex, allows for a more robust treatment of QBH. Furthermore, this representation broadens the range of applications, as will be described in the next subsection.

Melody detection in polyphonic audio is then the main subject of the present dissertation. Although we have highlighted its relevance to QBM, the utility of melody detection is by no means limited to this application. Other possibilities are described in the following paragraphs.

1.1.3. Other Applications of Melody Detection

Besides QBM, melody detection has applications in areas such as automatic music transcription, and melody transcription in particular. Automatic music transcription is rather time-consuming, error-prone and iterative, requiring in addition specialized skills. Thus, composers and music professionals could gain from automatic music transcription systems since these would free them for other more creative jobs. In this context, melody transcription, which could be viewed as a subset of full music transcription, is of particular interest, given the role played by melody in music (as will be described in the next chapter). In fact, users are often especially interested in the melodic part for musical composition purposes (e.g., for creating different versions of known songs), for learning or for copyright issues.

With respect to music education and training, automatic transcription system could assist students in their transcription proficiency. Such computer-aided education and training applications could automatically correct and evaluate users' results, keep track of progress, give information on the most common types of errors and so on.

Likewise, melody detection also opens possibilities for performance and expressiveness analysis. In effect, by comparing the written and the executed score, much information can be gathered as to the precision and style of music performers. Moreover, melody detection would too be useful in the development of computer tools for tasks such as music improvisation.

Plagiarism detection could gain from melody transcription as well. Indeed, authors would have the possibility of automatically comparing their copyrighted songs with new songs (or, conversely, their new works with existing pieces) based on melodic similarity measures [Grachten *et al.,* 2002], just like queries are matched to melodies in QBM.

As for music analysis, the melodic part contains useful information for the detection of motives and themes. Hence, its automatic transcription could support this task.

Concerning metadata applications, the implementation of the melody descriptors defined in the MPEG-7 standard [MPEG-7, 2004] would also benefit from algorithms for melody detection in polyphonic audio.

Other possible applications are offered in the field of music libraries, where it is often necessary to extract melodic descriptions directly from audio files.

The reasons pointed out to justify the need for automatic melody detection systems could be the subject of some criticism based on the fact that, nowadays, music is usually recorded in a multi-track fashion. In this way, monophonic analysis could be conducted on the melody track. This possibility would open new perspectives for most of the cited applications, namely automatic music transcription, provided that access to that data was ensured. Going even farther, multi-tags could be directly supplied by music editors. However, even if any of these procedures was always followed from this time forth and a concerted policy was agreed upon between the main editors and music retailers, huge amounts of recorded music still needed to be processed. Furthermore, it is not clear whether end-users would have indiscriminate access to such multi-track or multi-tag recordings. At least for the near future, we should expect music to be delivered as a mixture, as happens today, rather than in separate tracks. Therefore, if, for example, a music student needed to transcribe a musical piece, he would either do it manually or with recourse to an automatic transcriber.

1.2. Objectives and Approaches

As referred to in the previous section, melody detection in polyphonic audio is the main

subject of this dissertation. Despite its various possibilities, most of the involved research problems are complex and still open. We certainly have many years ahead before sufficiently robust, accurate and efficient melody detection algorithms become available.

Polyphonic musical signals can be converted into symbolic formats either manually or, ideally, automatically. Manual conversion requires obviously substantial man-work and specific skills. Moreover, this is a subjective and error-prone activity. On the other hand, analyzing polyphonic musical waveforms is a rather complex job since such signals are typically multi-timbral, having many different types of instruments playing concurrently, with severe spectral interference between each other.

Previous work on the extraction of symbolic representations from musical audio has concentrated especially on full music transcription. This demands accurate multi-pitch detection for the extraction of all (and nothing but) the fundamental frequencies (F0) present in a given song, besides requiring sound source separation in order to allocate each note to the respective instrument. However, the existing methodologies towards the mentioned problems are neither sufficiently general nor accurate. Namely, pitch detection accuracy decreases considerably as the number of sound sources increases. For that reason, some systems narrow the scope of the problem by imposing several constraints on the musical material, for example on the maximum number and type of simultaneous instruments or musical style.

In this way, we follow the pragmatic approach of putting the focus on the melody[16], no matter what other sources might be present. Rather than performing polyphonic pitch detection and full source separation, we propose a multi-stage mechanism complying with the principles of figure-ground organization (Section 2.2.2).

Thus, we implement a particular multi-pitch detection methodology, where, instead of aiming to capture all the pitches in each time frame, we only select the most relevant ones for melody detection. These are assumed to be the most salient[17] pitches. Then, we explicitly identify musical notes by creating pitch trajectories for the obtained pitch candidates, after which temporal trajectory segmentation is conducted with the purpose of separating all the individual notes contained in each track. In addition, we deal with the problem of ghost harmonically-related notes by incorporating perceptual cues of sound organization into our model, mimicking the human auditory system to some extent. As to the identification of the notes comprising the melody, we base our strategy on two main assumptions that we designate as the "salience principle" and the "melodic

[16] From this point forth, whenever we say melody we are actually referring to the "main melody" in a musical ensemble. The very concept of melody can be somewhat ambiguous. Hence, we propose a definition that suits well the context of our work and takes away possible ambiguities (Section 2.3).

[17] Throughout this document, we will employ the term "salience" to designate the (approximate) intensity (or energy) of notes, pitches or peaks. This term is preferred since it may equally well denote energies or probabilities. For example, a pitch may be salient either because its energy or its probability (regardless of how it is computed) is high.

smoothness principle". By the salience principle, we assume that the melodic notes have, in general, a higher intensity in the mixture (although this is not always the case). As for the melodic smoothness principle, we exploit the fact that pitch intervals between successive notes tend to be small. Finally, false notes in the obtained melody are deleted by discarding the ones that correspond to abrupt salience or duration reductions and by performing note clustering to further separate true melody notes from false positives.

The success of any melody detection system relies strongly on the efficacy of all its constituent modules. Namely, since melody has a lot to do with pitch, accurate pitch detection is a primary objective of our work. Therefore, we study both single and multi-pitch extraction algorithms in order to come up with a well-motivated choice of the most adequate method.

Moreover, we want to explicitly identify musical notes, contrariwise to most existing systems, which are mainly concerned with the extraction of predominant-pitch lines. Thus, musical notes must be precisely characterized, especially in terms of their timings and MIDI note numbers, for which the reliable creation of pitch trajectories is crucial. Hence, note determination is also an important objective of our work.

After acquiring a set of musical notes, we have to identify the ones that convey the melody, which is the ultimate objective of our project. In effect, regardless of how reliable pitch detection and note determination might be, the final goal is not achieved unless the notes bearing the melody are correctly extracted. In other words, accurate melody extraction depends on the reliable detection of musical notes, which, in turn, requires accurate pitch detection. Meaningful overall results are only possible if basic features are consistently extracted and properly integrated into a unified corpus.

1.3. Main Contributions

In this section, we summarize the main contributions of this work and list the set of publications that originated from the research carried out.

1.3.1. Pitch Detection

Our system starts with a melody-oriented pitch detection algorithm where an auditory-model-based pitch detector (AMPD) is employed and extended for the selection of multiple pitches. One of our basis assumptions is that melodic notes are usually salient in polyphonic mixtures, and so the strategy of selecting a few of the most intense F0s in each frame normally leads to positive results. However, in songs with low signal-to-noise ratio, peak masking occurs more prominently, mostly due to percussive sounds. Clarifying this point, for the purposes of this work, we consider everything that is not part of

the melody, i.e., all sorts of accompaniments, either pitched or percussive, as noise. Thus, we define the signal-to-noise ratio (SNR) as the relation between the intensity of the melodic instrument and the intensity of the background.

Experiments were also conducted towards frame-wise percussion elimination, but the outcome was somewhat unsatisfactory. Indeed, this is a complex task that needs further attention in the future. Anyway, the problem was attenuated by the allowance of trajectory inactivity in the construction of pitch tracks, which permits the restoration of undetected F0s.

Not many original developments are offered in this module. The main contributions pertain to the proposal, analysis and validation of a melody-oriented pitch detection scheme (which, basically, selects the highest peak candidates in each frame) and to a comparative evaluation of representative pitch detectors, employed in a musical context.

1.3.2. From Pitches to Notes

Unlike most other melody extraction systems, we explicitly identify musical notes, characterized by specific temporal boundaries, MIDI note numbers and intensity-related levels, storing as well the exact frequency values, which might be necessary for the analysis of performance dynamics such as vibrato[18], tremolo[19], glissando[20] or legato[21].

In this module, several novel contributions are provided. Namely, the adopted peak continuation approach (proposed by another author) is adapted and extended with a look-ahead procedure, intended to deal with peak competition between tracks, and with the reassignment of lost peak candidates to reduce track sparseness. Moreover, the derived pitch tracks may contain more than one single note and, therefore, must be segmented. The devised track segmentation mechanism is novel, except for the use (and adaptation) of an onset detector previously submitted by other researchers.

The accomplished results, despite showing that there is room for improvement, are positive. The main shortcomings of the algorithm come from its reliance on the definition of a minimum note duration, as well as from the current limitations of onset detection methods in polyphonic contexts. The former problem placed obstacles on the segmentation of pitch tracks with extreme vibrato, such as in opera pieces. The latter gave rise to difficulties on the accurate segmentation of consecutive notes at the same pitch.

---

[18] Periodic changes in the pitch of a tone, typical in opera singers (a.k.a. frequency modulation).

[19] Periodic changes in the intensity of a tone, typical in opera singers (a.k.a. amplitude modulation).

[20] Frequency slide in the attack of a note.

[21] Performing style where notes are smoothly "connected" without any perceptible break between them.

1.3.3. Identification of Melodic Notes

As a consequence of the employed multi-pitch detection scheme, several notes are created, among which the melody must be identified. This is not a trivial task since several features of auditory organization influence the perception of melody by humans, for instance in terms of the pitch, timbre and intensity content in musical ensembles. To this end, we have exploited aspects of intensity, where the most salient notes at each time are first selected, and frequency proximity, where the initially obtained melodic contour is smoothed out. All the developments in this module constitute novel contributions.

The obtained results were quite satisfactory. However, in sound signals where many salient non-melodic notes are present, e.g., musical pieces with low SNR (according to our previous characterization), the melody-smoothing algorithm had more difficulties in replacing the erroneous notes with the melodic ones. In fact, long smooth regions are validated, no matter whether they contain a high number of incorrect notes or not.

We have also devised a method for elimination of ghost harmonically-related notes. Here, we exploit principles of auditory organization, namely harmonicity and common fate. Most of the contributions in this task are original, except for the utilization (and adaptation) of a common modulation measure, proposed by other authors.

Additionally, we tackled the problem of false positives. As expected, this proved to be a very challenging task, and consequently only slight improvements were achieved. Spurious accompaniment notes that appear for brief moments during pauses between melodic notes were tentatively resolved by avoiding abrupt salience and duration transitions between consecutive notes. Then, note clustering was applied so as to separate accompaniment notes that are selected when the lead instrument stops. Nevertheless, note clustering lacked robustness. Indeed, despite overall improvements, the accuracy actually decayed in a few song excerpts. Moreover, the best feature set varied from excerpt to excerpt, which puts some difficulties in terms of its utilization in a general framework.

The approach for elimination of spurious accompaniment notes is also novel. With regard to note clustering, this was inspired by another work on melody detection. In the same way, the mechanism for feature extraction is based on previous research on instrument identification. In any case, the overall combined strategy is somewhat novel.

1.3.4. List of Publications

The main contributions of this project are summarized in the following publications:

(P1) Paiva R. P., Mendes T. and Cardoso A. (2004). "A Methodology for Detection of Melody in Polyphonic Musical Signals", *Proceedings of the 116th Audio Engineering*

Society Convention – AES116, Berlin, Germany.

(P2) Paiva R. P., Mendes T. and Cardoso A. (2005). "An Auditory Model Based Approach for Melody Detection in Polyphonic Musical Recordings", U. K. Wiil (ed.) *Computer Music Modeling and Retrieval – CMMR 2004*, Esbjerg, Denmark, *Lecture Notes in Computer Science*, vol. 3310, pp. 21-40.

(P3) Paiva R. P., Mendes T. and Cardoso A. (2005). "Segmentation of Pitch Tracks for Melody Detection in Polyphonic Audio", *Proceedings of the European Signal Processing Conference – EUSIPCO'2005*, Antalya, Turkey.

(P4) Paiva R. P., Mendes T. and Cardoso A. (2005). "On the Definition of Musical Notes from Pitch Tracks for Melody Detection in Polyphonic Recordings", *Proceedings of the International Conference on Digital Audio Effects – DAFx'05*, Madrid, Spain.

(P5) Paiva R. P., Mendes T. and Cardoso A. (2005). "Exploiting Melodic Smoothness for Melody Detection in Polyphonic Audio", *Proceedings of the International Computer Music Conference – ICMC'2005*, Barcelona, Spain.

(P6) Paiva R. P., Mendes T. and Cardoso A. (2005). "On the Detection of Melody Notes in Polyphonic Audio", *Proceedings of the International Conference on Music Information Retrieval – ISMIR'2005*, London, UK.

(P7) Paiva R. P., Mendes T. and Cardoso A. (2006). "Melody Detection in Polyphonic Musical Signals: Exploiting Perceptual Rules, Note Salience and Melodic Smoothness", *Computer Music Journal*, Vol. 30, No. 4 (to appear).

In the first article, we presented the initial system, with preliminary pitch detection, note identification and melody selection modules (publication P1). In this first attempt, a harmonic analysis scheme based on the Short-Time Fourier Transform (STFT) was implemented. The attained results were not convincing and, thus, an AMPD was employed in our second paper (publication P2).

The mechanism for note identification is described in publications P3 and P4. The latter is an extended version of the former, whereas, besides a more detailed description, a tuning compensation strategy is suggested.

The melody smoothing algorithm and the methodology for elimination of false positives were introduced in publications P5 and P6, respectively.

Finally, the overall system is described in a journal paper (publication P7), with particular accent on the melody identification stage. In addition, a more extensive examination of experimental results was fulfilled.

Besides attempting to validate our work by the usual peer review process in conference and journal papers, we have also participated in two melody extraction evaluations,

held at the 2004 and 2005 ISMIR conferences, which aimed to provide a quantitative comparison of different approaches. The conducted evaluations led to the following publications:

(TR1) Gómez E., Streich S., Ong B., Paiva R. P., Tappert S., Batke J.-M., Poliner G., Ellis D. and Bello J. P. (2006). *A Quantitative Comparison of Different Approaches for Melody Extraction from Polyphonic Audio Recordings*, Technical Report, Music Technology Group, Pompeu Fabra University, Spain.

(U1) Paiva R. P. (2005). "An Algorithm for Melody Detection in Polyphonic Recordings", *Proceedings of the Music Information Retrieval Exchange – MIREX'2005*, URL: http://www.music-ir.org/evaluation/mirex-results/articles/melody/paiva.pdf.

The melody extraction evaluation that took place as part of the Audio Description Contest (under the framework of ISMIR'2004) is described in the technical report (TR1), from the Music Technology Group of Pompeu Fabra University. Namely, the participating algorithms, ground truth data and evaluation metrics are presented, culminating in an experimental comparative analysis of the different methods.

As for the 2005 evaluation, a brief summary of our system is offered in paper (U1), published in the unreviewed online proceedings of the Music Information Retrieval Evaluation eXchange (MIREX'2005).

1.4. Outline of the Dissertation

We tackle the problem of melody detection in polyphonic audio following a multi-stage scheme, where a number of rule-based systems, inspired on principles from auditory physiology, perceptual theory and musical practice, are proposed. Our methodology comprises three main modules:

i) pitch detection;

ii) conversion of pitch sequences into musical notes (with precise temporal boundaries and pitches);

iii) identification of melodic notes.

The organization of this dissertation reflects the modularity of our approach, where each block is described in detail in a separate chapter. Two introductory chapters precede these more technical chapters.

Introductory Chapters: Chapters 1 and 2

Chapter 1

This is the current chapter, where we summarize the main motivations, objectives and contributions of this research project.

Chapter 2

Before describing the devised system, an overview of melody detection in polyphonic audio is provided in Chapter 2. We start with a brief introduction to music information retrieval, with particular focus on its audio branch, i.e., music content analysis.

The topic of music content analysis and listening is then further developed, where the most relevant perceptual and cognitive issues pertaining to it are presented, emphasizing aspects of melody perception.

Based on this discussion, and motivated by the context and assumptions of our work, we propose a definition of melody that suits our goals and context.

After that, we offer an overview of some of the work on closely related topics such as automatic music transcription and review the state of the art on the specific melody detection problem.

We then summarize our system, describing its main constituent modules, the underlying assumptions and the strategies pursued.

Finally, the employed evaluation databases and metrics are described.

Technical Chapters: Chapters 3, 4 and 5

The overview is followed by three more technical chapters, which contain the main contributions of this dissertation. Each of these chapters starts with an introductory section with the global idea of the problem under study and the respective research status, after which our approach is described. Each (sub-)module is summarized in a (sub-)section titled "Putting It All Together", where the entire method is condensed in algorithmic form. Finally, experimental results are analyzed, the main experienced difficulties are discussed and suggestions for improvement are pointed out.

Chapter 3

In accordance with these lines, pitch detection is introduced in Chapter 3. Pitch is the main low-level signal feature in melody detection tasks. Substantial work has been devoted to this topic throughout the years, especially in the context of monophonic speech processing. More recently, pitch detection algorithms have been proposed to deal specifically with musical signals, both in monophonic and polyphonic contexts.

In this chapter, the employed AMPD is described in detail and compared to other evaluated algorithms, namely a method employing simple autocorrelation, another one based on spectrum autocorrelation, one more using the STFT and another one following a probabilistic approach. From the conducted study, the AMPD is selected, as a result of its better accuracy.

In addition, mechanisms for pre and post-processing of musical signals are investigated (e.g., RASTA, i.e., RelAtive SpecTrAl and enhancement of the summary autocorrelation function).

Chapter 4

The note is the fundamental representational symbol in Western music notation. Even so, the accurate identification of musical notes, regarded as musicological units with dynamic nature, is somewhat overlooked in automatic music transcription research. Therefore, in Chapter 4 we devise a method for quantizing the temporal sequences of detected pitches into discrete note symbols, characterized by precise timings and MIDI note numbers, besides coping with typical dynamics and performing styles.

Our algorithm starts with the construction of a set of pitch tracks, formed by connecting pitch candidates with similar frequency values in consecutive frames. The objective is to find regions of stable pitches, which indicate the presence of musical notes.

Since the derived trajectories may contain more than one note, temporal segmentation must be carried out. This is accomplished in two steps, making use of the pitch and salience contours of each track, i.e., frequency and salience-based segmentation. In frequency-based track segmentation, the goal is to separate all notes of different pitches that might be included in the same trajectory, handling glissando, legato, vibrato and frequency modulation in general. Concerning salience-based segmentation, the objective is to separate consecutive notes at the same pitch, which may have been incorrectly interpreted as forming one single note.

Chapter 5

In Chapter 5, we describe our efforts towards the identification of melodic notes in a mixture. Our strategy is grounded on the assumptions that the main melodic line often stands out in the mixture and that melodic contours are usually smooth in terms of pitch intervals.

Moreover, the problem of accompaniment notes present in the obtained melody is dealt with by excluding the ones that correspond to abrupt salience or duration reductions and by performing note clustering to further discriminate the melody from the accompaniment.

The algorithm for eliminating ghost harmonically-related notes is also discussed in this chapter.

Conclusion: Chapter 6

Finally, we sum up the main conclusions of this work and point out possible directions for future research. This discussion is based on the main encountered difficulties and on the relation of this project to music information retrieval.

Bibliography

All cited references are listed here (journal and conference papers, books, book chapters, theses, online references, etc.).

Appendices

In Appendix A, we describe the other evaluated pitch detection methods. As will be seen, the auditory-model-based pitch detector was preferred over these, mainly due to its better overall accuracy, the reason why these were moved to an appendix. In any case, comparative results are presented and discussed in Section 3.6.

In Appendix B, the song excerpts used in the evaluation of our work (and listed in Chapter 2) are qualitatively characterized in terms of category, solo type, polyphonic complexity, signal-to-noise ratio, duration, number of melody notes and other peculiarities specific to each of them.

Chapter 2

Melody Detection: Context and Overview

> *"It is melody that enables us to distinguish one work from another. It is melody that human beings are innately able to reproduce by singing, humming and whistling. It is melody that makes music memorable: we are likely to recall a tune long after we have forgotten its text."*

Eleanor Selfridge-Field, "Conceptual and Representational Issues in Melodic Comparison, pp. 4", 1998

Melody detection is presently deserving an increasing interest by the Music Information Retrieval research community. In fact, automatic melody extraction from polyphonic audio fills an important gap in query-by-melody systems, besides having other applications in areas such as music education, composition or plagiarism detection. Given the relevance of melody detection to MIR, the main aspects related to both subjects are discussed in this chapter.

Section 2.1. Music Information Retrieval (MIR)

Under this perspective, we start with a brief introduction to music information retrieval, with particular accent on its audio branch, i.e., music content analysis.

Section 2.2. Content Analysis and Music Listening

Next, we further develop the topics of music content analysis and listening, discussing some of the perceptual and cognitive issues involved and highlighting matters of melody perception.

Section 2.3. Melody Definition

From the previous point, it becomes apparent that music listening in general, and melody perception in particular, are inherently subjective processes. As a consequence, characterizing melody in clear and unequivocal terms is not immediate. Therefore, we suggest a definition of melody that suits our goals and context of analysis.

Section 2.4. Melody Detection in MIR Research

We then address the role of melody detection in MIR research, offering an overview of some of the existing work on closely related matters such as automatic music transcription and reviewing the state of the art on the specific melody extraction problem.

Section 2.5. Overview of the Proposed Melody Detection System

After introducing the state of the art, we sketch our melody detection system, describing its main constituent modules, the underlying assumptions and the strategies pursued.

Section 2.6. Test Collections and Evaluation Procedures

Finally, we describe the efforts undertaken by researchers in this field towards the acquisition of ground truth data and the development of evaluation metrics.

2.1. Music Information Retrieval (MIR)

Music Information Retrieval is an emergent and promising research area that has evolved from the necessity to manage huge collections of digital music for "preservation, access, research and other uses" [Futrelle and Downie, 2003].

Despite the surge of interest in recent years, the idea of music information retrieval dates back to the 1960's, where the potential of applying automatic information retrieval techniques to music was recognized [Kassler, 1966] (cited in [Uitdenbogerd *et al.*, 2000]). Moreover, we can look at incipit and theme indexes, e.g., Harold Barlow and Sam Morganstern's dictionary of musical themes [Barlow and Morganstern, 1948] (cited in [Uitdenbogerd *et al.*, 2000]), as the precursors of computer-based MIR.

The current ever-increasing awareness given to MIR research is a direct consequence of the explosion of the EMD industry, promoted by the generalized access to musical materials in digital form (with particular emphasis on compact audio formats with CD or near CD quality, such as mp3), widespread Internet availability, with increasing bandwidth at reduced costs in domestic connections, and by the creation of online

peer-to-peer services such as Napster, Shareazaa or Kazaa.

During the 1990's, music information retrieval became a topic of growing interest, for example in areas such as query-by-humming. This trend has continued and these days MIR has established itself as an important interdisciplinary research field of its own, with dedicated conferences and research laboratories spread all over the world. Particularly, the MIR community gathered for the first time in 2000, during the 1st International Symposium on Music Information Retrieval – ISMIR'2000[22], and from that time forth with a yearly periodicity.

2.1.1. MIR applications

As referred to in the previous chapter, MIR is unquestionably an area with tremendous application potential. Subjects such as automatic music classification and feature extraction, audio fingerprinting, music recommendation, automatic music transcription, melody detection, song database indexing, music representations or user interface design, are matters of active research.

In relation to platforms for EMD, music web crawlers, which "traverse the web and index music-related files" [Huron, 2000], open several possibilities. In addition, huge music databases would benefit from automatic classification tools, both in content labeling and updating, as well as from mechanisms for content-based retrieval. This also applies to multimedia databases and operating systems.

Similarity-based retrieval tools have also a vast potential, e.g., in automatic playlist generation [Pauws and Wijdeven, 2005; Alghoniemy and Tewfik, 2000] and music recommendation [Celma *et al.,* 2005; Vembu and Baumann, 2004].

Besides the possibilities for EMD, platforms for education and training can gain as well from the conducted efforts. For example, mechanisms for automatic music transcription (e.g., [Ryynänen and Klapuri, 2005a; Klapuri, 2004; Bello, 2003; Sterian, 1999; Martin, 1996; Kashino *et al.,* 1995]) might simplify tasks such as manual transcription, music composition, music analysis or evaluation of musical performances (for instance, by the examination of the employed dynamics or the automatic comparison of the written score with the executed one). Furthermore, professional composers might find useful tools that support plagiarism detection.

Digital music libraries are also an interesting application field of MIR research [Dunn, 2000; Fingerhut, 1999]. One example of this is the VARIATIONS project [Dunn, 2000], afterwards upgraded to Variations2. The referred project attends to both

[22] For historical reasons, the acronym ISMIR was maintained after changing the event from a "Symposium" to a "Conference" in 2002, i.e., ISMIR was kept instead of being renamed to ICMIR. For more information on the early history of ISMIR, see [Byrd, 2002].

technical issues, for example, content-based information retrieval, and educational ones such as learning activities for music instruction and evaluation of learning impact, supported by the library.

Additionally, audio editors or audio browsers would become more "intelligent" with the incorporation of mechanisms for automatic indexing of music/audio files [Wold *et al.*, 1996].

As for the advertising and cinema industries, tools for mood-based music retrieval would certainly be beneficial, since it is often necessary to search for songs that induce a certain mood to the intended audience [Huron, 2000].

Video indexing and searching could gain as well from music content analysis and, more generally, from audio content analysis. In effect, rather than solely inspecting image frames, it is known that the analysis of the audio stream can support the detection of scene transitions, essential to video indexing and segmentation. For example, romantic scenes (love inspiring song) or violence (shots, screams) can be detected by looking only at audio information [Pfeiffer *et al.*, 1996].

2.1.2. MIR Representations and Research Areas

In conformity with the state of the art, Joe Futrelle and Stephen Downie proposed a categorization of the major MIR research themes, as well as their fundamental topics and investigation needs [Futrelle and Downie, 2003]. Mostly matters of basic research were spotted, including subjects such as representation, indexing, retrieval, user interface design, music recommendation, audio compression, feature detection, classification and machine learning, musical analysis, summarization, metadata, users studies, intellectual property rights, perception, epistemology and ontology.

These research themes can be roughly grouped according to the kind of music representation they employ [Futrelle and Downie, 2003]. Namely, some of the identified areas address topics such as melodic matching, theme extraction or musical analysis, thus requiring symbolic music representations, e.g., scores or MIDI. These subjects fall under the *symbolic MIR* category. Other areas deal mostly with audio recordings or streaming audio, focusing on tasks such as automatic transcription, QBE or classification. This branch is denominated *audio MIR*. Visual representations are also adopted, for example in optical music recognition, giving rise to the *visual MIR* group. Finally, *metadata MIR* tackles research matters that require metadata representations, such as the cataloguing process in digital libraries.

One important category of MIR research is the one termed audio MIR, which deals with music information retrieval in audio signals. Our work falls under this class, as it involves aspects of MIR more closely related to *music content analysis* (MCA), particularly

melody detection in polyphonic audio. In any case, the motivations of our research grasp features of symbolic MIR as well, since we obtain a set of musical notes that might be used, for example, in melody matching.

Music content analysis focuses on the use of computers to examine recorded or performed music. In other words, computers act as music-listening machines, although they may or may not aim to mimic the operations conducted in the human auditory system. For this reason, designations like *Music Listening* [Scheirer, 2000] or *Music Scene Analysis* [Kashino *et al.,* 1995] are also used in the context of MCA. Anyway, in our opinion, the first denomination has a strong connotation to physiological and perceptual issues, whereas the second one relates to the recognition of the music producing objects in an "auditory scene", as will be seen in Section 2.2. Therefore, we prefer the term "music content analysis" since it seems less restrictive and, hence, applicable to a broader range of problems (this is based on similar arguments presented in [Tzanetakis, 2002, pp. 4]).

As far as musical audio is concerned, content can be looked upon both as the *explicit* audio information bore in a signal and the *implicit* information associated with it, namely its structural, rhythmic, instrumental and melodic characteristics [Gómez, 2002, pp. 9]. For instance, the number of instruments a song, the melodic line in a given part or the musical key are examples of content information.

The need for easy and meaningful interaction with this kind of data has prompted research into techniques for the automatic description and handling of audio. Several of these methodologies attend to topics such as automatic transcription, rhythm and melody characterization, instrument recognition and genre or artist classification. Among these, melody plays a major role in the context of MCA. Even so, melody extraction from polyphonic musical recordings has received far less research attention than other MCA problems, such as tempo, beat and meter estimation.

From a different perspective, music content analysis can also be viewed as a branch of the broader *Multimedia Content Analysis* field, which "refers to the computerized understanding of the semantic meanings of a multimedia document, such as a video sequence with an accompanying audio track" [Wang *et al.,* 2000]. In music content analysis, the accent is placed on the audio stream, namely in the analysis of musical signals.

Most mechanisms for MCA are still in embryonic stages, mainly due to the little attention conferred to this topic up until recently. In reality, despite the importance of the audio stream in multimedia systems, "most research and development work - such as videoconferencing systems, video-on-demand and multimedia databases - has focused on the video stream. In each of these areas, music is of minor interest. The audio track in videoconferencing consists of speech only. Video-on-demand typically contains a mixture of speech, noises and music. Often it's interleaved with the audio stream; little work has been done on the extraction and specific processing of the music component. In the field of multimedia databases, research and development work emphasizes the indexing

and retrieval of still images and video rather than addressing audio issues" [Effelsberg, 1998].

Significantly less research work has been carried out in the audio stream of multimedia systems, in comparison to the amount of work devoted to the textual, image and video streams. Furthermore, the bulk of research in audio content analysis has concentrated on speech processing (e.g., [Rabiner and Juang, 1993]). Indeed, the roots of research in this area are in speech processing and recognition [Tzanetakis, 2002, pp. 14]. Only recently, researchers have started to take advantage of the potential of music in tasks such as video indexing or semantic analysis, e.g., [Wang *et al.*, 2000; Minami *et al.*, 1998].

2.1.3. MIR Methodological Needs

Research in MIR is still in its infancy, with several interesting but complex and open problems. In fact, the existing approaches towards the abovementioned challenges are still insufficient in terms of accuracy, generality and robustness.

Moreover, MIR is a markedly inter-disciplinary field, owing background from established areas such as information retrieval, musicology, psychoacoustics, music cognition, computer music, digital signal processing, automatic speech recognition, statistics, artificial intelligence, human-computer interaction, library science, publishing or law.

Fundamental differences exist between its diverse disciplinary communities, as generally happens in any emergent multi-disciplinary subject. Nonetheless, such divergences must be worked out and synergies between them should be exploited. Namely, it is essential that the different communities "articulate a common research agenda or agree on methodological principles and metrics of success" [Futrelle and Downie, 2003]. In effect, MIR's richness and novelty make it a profitable terrain for research innovation but, at the same time, lead to a lack of consensus concerning methodological standards.

Uniformity in evaluation strategies is an important, until recently, unfulfilled requirement in MIR research: there was no universal agreement on standard benchmark problems and evaluation metrics, which researchers might use for comparing their respective approaches. Fortunately, some efforts are now being conducted in this respect, as will be discussed in Section 2.1.4.

As for users' needs, a formal and extensive examination is also lacking. Presently, a few research topics have been considerably emphasized without rigorous studies regarding the preferences of users. Actually, ad-hoc arguments are often proposed to motivate research on some matter, without a clear and profound assessment of users' needs. Particularly, QBH is one of the predominant retrieval paradigms in MIR, based on arguments that it is natural and intuitive, although there are no objective studies that sustain

the widespread idea that this is the modality preferred by users[23] [Futrelle and Downie, 2003]. On the other hand, researchers need that diversified platforms be developed in order to support the study of users' behaviors. As Futrelle and Downie refer [Futrelle and Downie, 2003], this is a chicken-and-egg problem: development must be grounded on user's needs but studies on the preferences and behaviors of users also depend on the development of techniques... In addition, meaningful studies about needs and preferences of users, as well as system evaluation, require large and consensual data collections.

Hence, the three issues pointed out (creation of standard data collections and evaluation metrics, analysis of users' needs and advances in new techniques and tools) should go alongside. In our opinion, a balanced position would be to go on with fundamental research without putting too much accent on overly specific topics. Only when substantial critical mass and sizeable and comprehensive test-beds are available will user studies become more significant and confirm or refute the continuation on specific research tracks. The kind of basic research we carry out in this dissertation keeps in with these lines.

2.1.4. MIR Evaluation Methodologies

Evaluating research on music information retrieval, namely in what regards the aspects more intimately related to human perception, is a complex endeavor due to the inherent subjectivity associated with those mechanisms. This in turn makes it difficult to come up with definitive, unambiguous and correct answers to some of the questions involved.

In order for the evaluation of research outcomes to be as accurate and objective as possible, experiments must be carefully designed, which frequently requires the realization of user studies. For example, in QBE and QBM, performance evaluation is usually measured by determining the proportion of relevant returned answers. With the purpose of defining relevancy, human evaluators typically look at each potential answer and judge accordingly. A number of users are asked to personally evaluate the results, which normally consist on counting the average number of similar songs in the first 5, 10 or 20 in the output list. This clearly subjective metric arises as a consequence of the fuzziness around the concept of similarity [Orpen and Huron, 1992]. Thus, a meaningful ground truth would require a number of human evaluators. For this reason, detailed evaluation reports are often missing because of the difficulties in conducting thorough end-user tests. Furthermore, in genre classification tasks, large data sets whose labeling is consen-

[23] In some informal conversations with people who use the Internet frequently to look for music, we have noticed that the generality reacts enthusiastically to the idea of QBH. However, this could be just a natural reaction to a cutting-edge technology. Nothing assures that the implementation of QBH platforms for real-world song databases would be a success. Other aspects, such as efficiency or scalability, would have a strong impact on its usefulness.

sual, as well as standard evaluation metrics, are crucial.

As referred to, up until recently there was no general agreement as to standard benchmark problems and evaluation metrics in MIR. Consequently, most of the obtained results are hard to compare with related work. Different authors use different testbeds and devise their own accuracy metrics, some of them based on questionable criteria. Thus, comparative studies are rare, since they would require complex systems to be implemented from scratch. Additionally, the data sets employed in some works are very small, which raises several questions on the significance of the results.

Fortunately, this problem is now receiving further attention. Particularly, this was the subject of the Workshop on the "Creation of Standardized Test Collections, Tasks and Metrics for Music Information Retrieval (MIR) and Music Digital Library (MDL) Evaluation", which took place during the Second Joint Conference on Digital Libraries (JCDL' 2002). This was also discussed in a panel held at ISMIR'2002, titled "Music Information Retrieval Evaluation Frameworks" [Downie, 2002].

More recently, the yearly organization of the Music Information Retrieval Evaluation eXchange, which started in 2004[24], has led to the definition of a set of databases devoted to different specific jobs (e.g., melody extraction, genre classification, drum detection, etc.), as well as uniform evaluation methodologies, which researchers are able to use for algorithm benchmarking. Moreover, a common platform for comparison of different approaches is set up, which fills a very important gap in MIR research.

Despite these initial and important efforts, much work is still needed. Speaking specifically of the research challenges involved in melody extraction, extensive and more comprehensive datasets are required. As will be discussed in Section 2.6, it is our opinion that the existing databases are small and their content is not sufficiently diverse.

2.2. Content Analysis and Music Listening

As referred to in Section 2.1.2, the musical content of a piece can be looked upon as the implicit information enclosed in it, namely its structural, rhythmic, instrumental and melodic characteristics. Examples of content information include the number of instruments, the melodic line conveyed in a given musical part or the key of a song.

In the development of tools for automatic music content analysis, a purely computational model, a physiological-perceptual scheme or a mix of both can be pursued. In the following, we will succinctly describe these paradigms, highlighting the perceptual issues

[24] The first worldwide MIR evaluation was held at the ISMIR'2004 conference. The performed "Audio Description Contest" was the precursor of the Music Information Retrieval Evaluation eXchange, formally set up in the following year. Anyway, in this document we designate the 2004 initiative as MIREX'2004.

entailed in the music-listening experience, regarding particularly its melodic aspects. The role played by higher-level cognitive processes is also briefly discussed.

2.2.1. Music Content Analysis Paradigms

In the computational (or black-box) approach, the focus is placed on the output results regardless of the method adopted to accomplish them: the goal is to construct a functioning machine, no matter what techniques are employed. Purely computational models are pragmatically goal-oriented, since the devised algorithms are less important than the outcome. This is a possible advantage given that, ideally, it is possible to attain performances that surpass the ones delivered by the human auditory system. For example, tasks such as automatic music transcription are very difficult for humans, as will be discussed; accurate computer tools could, therefore, outperform human transcribers. The drawback is that, so far, purely black-box schemes are a long way behind the accuracy of humans in hearing-related problems. Furthermore, with such algorithms, e.g., neural networks, we are not able to grasp what the computer is actually doing.

On the other hand, in the physiological-perceptual (or clear-box) paradigm the objective is to perceive what humans do and in the ways they do, i.e., the goal is to model exactly the human auditory system. This is a potential benefit since this is up until now the only working mechanism available. Consequently, it would be wise to comply with a strategy identical to the one of a working "machine". Besides, by building up artificially hearing instruments, we can gain insights on how the "real-thing" works. The downside is that, though many aspects of the physiology of hearing have been investigated and are well-known and relatively consensual (mostly in the peripheral regions of the human auditory system), several facets of auditory perception are only superficially understood (these are mostly located in the brain's central nervous system and, unlike the others, can be studied only indirectly). Additionally, computer models that mimic the human behavior tend to be computationally expensive.

Both paradigms have advantages and shortcomings. Either way, in the current state of affairs a lot could be gained from a more profound understanding of the mechanisms that govern auditory perception in humans, since no other system has proved to perform any better thus far. In our opinion, a compromise between the two possibilities should be the best option: computational models could be developed by exploiting both black-box analysis tools and available knowledge about the properties of the human auditory system. Hence, the available knowledge pertaining to its functioning should, at least, be taken into consideration. As will be seen later on, our method adopts this so-called "gray-box" approach to some extent. Therefore, in the next sections we describe some of the features involved in perceptual sound organization, particularly the ones related to music and melody.

2.2.2. Music and Melody Perception

Human listening comprises many layers of information analysis and processing. These include the treatment of low-level auditory stimuli in the ears and their organization in the brain, which is influenced by higher-level factors such as memory, experience and context information. Moreover, listening is also affected by several variables, interacting, cooperating and competing with each other, which makes it rather complex. In any case, in the context of our research, we skim the surface of some of the underlying procedures, knowing in advance that such a summary will miss many important aspects, thus lacking accuracy and completeness. A comprehensive study of the problem can be found in [Bregman, 1990; Handel, 1989].

The human ear is responsible for the primary auditory sensations, which will then be passed on to the brain for interpretation. Without entering into too much detail on the physiology of the ear, it can be briefly summarized as follows (more information will be provided in Section 3.3): the sound that reaches the ears is converted into nervous impulses by the hair cells in the cochlea, due to the movements of the basilar membrane; the firings of a nerve connected to a particular hair cell show band-pass response to the sound, and so the cochlea acts as frequency analyzer; furthermore, the density of such firings depends on the intensity of the input signal.

There is a wide consensus regarding these basic levels of processing. However, much more debate arises in relation to the way the human brain actually organizes the basic auditory stimuli it receives. One landmark contribution to the understanding of perceptual sound organization in the human auditory system is Albert Bregman's "Auditory Scene Analysis" [Bregman, 1990]. In this book, the author extensively documents most of the available knowledge on the mechanisms involved in human listening.

A. Auditory Scene Analysis

Auditory scene analysis is described as the process by which humans use sound to create a "picture" of the sonic world around them, i.e., to build mental descriptions of the *auditory scene*. In his work, Bregman summed up the existing knowledge on the subject, performed a myriad of psychoacoustic experiments and proposed several theories about the ways humans perceive auditory information.

The idea of auditory scene analysis is usually presented with recourse to the classical "cocktail party effect" [Handel, 1989, pp. 189]. This is related to the idea of selective attention, by which we are able to focus on one conversation in the midst of other dialogues and noise, e.g., background music, pouring drinks, people laughing, etc.

Albert Bregman metaphorically established a curious parallel between the audio waves in auditory scene analysis and water waves in a lake [Bregman, 1990, pp. 5]: "Imagine that you are on the edge of a lake an a friend challenges you to play a game. The

game is this: Your friend digs two narrow channels up from the side of the lake. Each is a few feet long and a few inches wide and they are spaced a few feet apart. Halfway up each one, your friend stretches a handkerchief and fastens it to the sides of the channel. As waves reach the side of the lake they travel up the channels and cause the two handkerchiefs to go in motion. You are allowed to look only to the handkerchiefs and from their motions to answer a series of questions: How many boats are there on the lake and where are they? Which is the most powerful one? Which one is closer? Is the wind blowing? Has any large object been dropped suddenly into the lake?"

In music content analysis, the processing mechanism, be it human or computational, performs a kind of auditory scene analysis, which we could term "music scene analysis", as previously referred to. Particularly, in this dissertation the "listener" must capture the melody in the midst of the surrounding accompaniment.

Melody perception is of special interest to our work. An important issue here is that, even though in its essence melody is nothing else than a succession of pitches in time, the auditory components that carry melody information somehow form an integrated corpus that is heard as a unity. No definitive answers on how this is conducted in the brain are available so far. This and other questions have long intrigued philosophers and scientists, who, ever since the ancient Greeks, have theorized about the modus operandi of the human auditory system.

Despite this interest, research on audio perception has received less attention than that on the visual domain. Nevertheless, psychological studies have given evidence that the human auditory and visual systems, despite their numerous differences, have several common processing elements. In fact, phenomena such as exclusive allocation or apparent motion occur both in visual and auditory processing. Thus, audio perception researchers have drawn from past research in human visual perception, as it supplied an important frame of reference for investigation and experimentation.

In the early part of the 20th century, a group of German psychologists, later known as Gestalt psychologists, suggested a number of principles that could account for the perception of visual information. They derived a set of laws that seemed to regulate the ways the human brain groups elements in a visual scene to represent shape or form at a larger scale than the individual elements. Interestingly, such laws seemed to apply also to the perception of auditory information [Bregman, 1990, pp. 18-28, 52].

B. Perceptual Units

The compositional analysis of acoustic waves is an inherently complex and ambiguous task. In effect, any waveform may contain superimposed information from different sources, as illustrated by the cocktail party example. In this way, the distinctiveness of each component is lost in the composite signal.

In order to unscramble the acoustic wave and capture the *auditory objects* enclosed in

it, the human auditory system uses relationships between the received sensory stimuli to partition the sound wave into such auditory objects. Bregman denominates them *auditory streams*, i.e., perceptual units that represent single events [Bregman, 1990, pp. 10]. For example, a sequence of similar frequencies may be combined to create the perception of a single musical tone.

It seems that such partitioning is governed by a number of fairly simple rules proposed by Gestalt psychologists. Generally speaking, Gestalt principles can be looked upon as rules of thumb that give hints on the high-level perception that results from certain low-level stimuli[25]. Such principles make use of the ideas of similarity, proximity, continuity or common fate [Handel, 1989, pp. 187; Bregman, 1990, 196-202]. Basically, the parts of an acoustic wave that are grouped into one perceptual unit are expected to be similar (e.g., in frequency, timbre and/or intensity), to be spatially or temporally close and to follow the same time trajectory regarding their frequency, intensity or position, complying with the law of Prägnanz. [Handel, 1989, pp. 187]. This law roughly says that "we try to experience things in as good a gestalt way as possible. In this sense, "good" can mean several things, such as regular, orderly, simplistic, symmetrical, etc."[26].

These organizational cues are employed in the grouping and segregation of both sequential and simultaneous information [Bregman, 1990, pp. 30]. Sequential integration is related to the organization of consecutive acoustic elements into a single auditory object. This can be accomplished at different levels, e.g., grouping together consecutive frequencies into one tone or combing a succession of notes into a melodic line. Simultaneous integration refers to the process of assembling distinct acoustic elements occurring at the same time but at different spectral or place locations into a single perceptual unit, e.g., the grouping of different harmonics in the perception of a single tone. Sequential and simultaneous integration are not mutually exclusive. Rather, they interact and compete with one another, much in the same way as the horizontal and vertical dimensions of written music do.

The perceived auditory objects do not always correspond to individual sounds. For example, a chord, formed by the combination of simultaneously sounding notes, may be perceptually interpreted as a single coherent auditory object. Additionally, an entire melodic phrase, consisting of a succession of several notes, usually forms an integrated unit. Indeed, as previously mentioned, human listening involves many layers of processing that combine sensory information according to diverse perceptual cues, as well as previous knowledge, experience or memory. In this way, at one level individual notes may correspond to individual objects, whereas at higher abstraction levels the succession of

---

[25] Gestalt is the German word for shape or form. This school of psychology "interprets phenomena as organized wholes rather than as aggregates of distinct parts, maintaining that the whole is greater than the sum of its parts" (as defined in Answers.com).

[26] Answers.com: http://www.answers.com/topic/gestalt-psychology.

notes may form an integrated corpus. In music, this hierarchical organization of units is carried out based on different time scales [Bregman, 1990, pp. 72].

C. Sequential Integration

The organization of sequential information relies on aspects such as proximity, similarity and continuity.

Proximity

This cue makes use of the hypothesis that sensory elements in close proximity tend to be grouped together as a unit. In terms of audition, proximity can be looked upon in two dimensions: time and frequency [Bregman, 1990, pp. 19, 58-67]. Thus, sensory elements close together in time and/or having close frequency values might have originated from the same physical source.

The importance of this cue is reflected in the fact that melodies tend to use small pitch intervals between consecutive notes. Violations of proximity have been used, for example, in the creation of auditory illusions such as fission. This is the case of so-called "virtual polyphonies", present in some Baroque compositions, e.g., by Bach and Teleman [Bregman, 1990, pp. 464]. There, a solo instrument alternating between a high and a low pitch register might lead to the perception of two distinct melodic lines played concurrently. This results from the tendency to perceive two streams when the speed of succession and frequency separation between consecutive notes is sufficient.

The principle of time-frequency proximity, as well as related musicological practices, motivated the development of our melody-smoothing algorithm (Section 5.4).

Similarity

In audition, elements that sound alike, i.e., have similar timbres, tend to be grouped together as a unit [Bregman, 1990, pp. 19; 92-127]. Hence, in a musical ensemble, the sounds from a given source are grouped together and separated from the sounds of other instruments. However, as we will discuss in Section 5.6, timbre is a somewhat vague concept, difficult to measure physically.

The auditory organization of physical stimuli also resorts to loudness similarity, though it is not as important for the perception of musical parts.

Grouping by similarity is influenced as well by temporal proximity. In effect, the likelihood that similar auditory elements are part of the same sonic event is increased by their temporal closeness.

Continuity

Human perception tends to continue contours whenever the elements follow a pat-

tern indicating a given direction. It is in this fashion that the human auditory system performs auditory restoration as a response to simultaneous masking. Simultaneous masking occurs when a loud sound covers a softer one, i.e., masks it. Nevertheless, if the softer sound is longer and can be heard both before and after the louder one, the former is often "heard through" the latter [Bregman, 1990, pp. 27, 133-136]. The inactivity time that is allowed in the construction of pitch tracks relates to this principle (Section 4.2).

Associated with this point is the phenomenon of perceptual restoration [Bregman, 1990, pp. 347]. Here, even if the softer sound is actually removed during the occurrence of the louder sound, it will still be perceived as an uninterrupted signal under some conditions, e.g., if some of the spectral content of the louder sound is similar to the one the softer sound would have if present [Bregman, 1990, pp. 349].

D. Simultaneous Integration

Harmonicity or common fate are involved as well on the combination and segregation of simultaneous information.

Harmonicity

It is generally accepted that the harmonic relations between spectral components are very important to sound fusion [Bregman, 1990, pp. 227-248]. Here, harmonically-related spectral partials, i.e., frequency components forming a pattern of (nearly) integer multiples of a common fundamental frequency, tend to be grouped together. For instance, the spectral components of the sounds produced by pitched musical instruments usually conform to a harmonic pattern that is exploited by the brain to fuse them into a single unit.

Common Fate

Sonic elements can also be fused if they "move" in the same way [Bregman, 1990, 248-292]. Hence, cues such as common onsets or endings, where components appear and disappear approximately at the same time, suggest that they might be part of the same sonic event, and so should be combined. Namely, onset synchrony seems particularly significant in the grouping of simultaneous sounds [Bregman, 1990, pp. 213-216].

The same indication is provided by common modulation, where elements have synchronized and parallel changes in frequency or intensity. This happens, for example, in vibrato, where the several harmonics of a tone show parallel frequency variations. On the other hand, harmonically-related elements with different frequency modulations are usually segregated [Bregman, 1990, pp. 255].

We take advantage of both common fate and harmonicity for merging of note candidates, as will be discussed in Section 5.2.

E. Tonal Fusion and Musical Notes

Simultaneous integration is not solely related to the grouping of harmonics, but also to the fusion of concurrent sounds stemming from different sources [Bregman, 1990, pp. 245]. Indeed, the perception of multiple simultaneous musical tones as single auditory objects plays an important role in music listening [Scheirer, 2000, pp. 30].

This is frequently explored in music orchestrations, which often "force" the fusion of sounds from different origins into one unified perceptual element. For example, "synchronous onset times and harmonic pitch relations are used to knit together sounds so that they are able to represent higher-level forms that could not be expressed by the atomic sounds separately" [Klapuri, 2004, pp. 9]. Such tonal fusion is even stronger when the number of shared harmonics is high. For example, the sounds produced by pipe organs perceptually fuse into one single percept, whose global properties are not merely the sum of the properties of the individual sounds.

In those situations, multiple sounds are intentionally bound together in order that a single, perceptually indivisible, auditory object is perceived. Albert Bregman coined the term *chimera* to describe the perceptual construct that results from the tonal fusion of the individual sonic elements [Bregman, 1990, pp. 5, 459].

Chimeric sounds that emerge, for example, by the fusion of different simultaneously played notes raises the question of whether or not musical notes correspond to perceptual constructs. In effect, in those cases musical notes do not seem to be individually perceived (at least without a conscious effort). Going farther in this direction, Eric Scheirer argues that, for most listeners, the perceptual objects created in the human auditory system generally have nothing to do with musical notes [Scheirer, 2000, pp. 67-69]. This point will be further discussed in Section 4.1.

F. Spatial Location

Besides the listed organization principles, simultaneous and sequential integration are both influenced by spatial location. Cues such as interaural delays are used in the perception of the physical origin of sounds [Bregman, 1990, 73-82, 293-311]. Hence, sounds perceived as arising from the same place also tend to be fused together. This rule is not utilized in our work, since we only use monaural recordings given that the melody can be easily identified even there.

G. Higher-Level Cognitive Aspects of Melody and Music Listening

In the previously described mechanisms, information flows essentially bottom-up. Briefly, sensory data coming from low-level acoustic signals is organized in accordance with Gestalt principles and passed afterwards to the higher-level processing elements in the brain for interpretation. Bregman refers to this as *primitive organization* [Bregman,

1990, pp. 38], in the sense that it corresponds to innate, unlearned, constraints.

However, it is generally accepted that listening also resorts to aspects dealing with previous knowledge, learning, memory or context. These govern voluntary attention, the creation of expectations and the resolution of perceptually ambiguous situations [Bregman, 1990, pp. 398]. For example, musical scales can be perceptually restored when one entire note is removed and replaced by a loud noise burst, based on the predictions drawn from higher-level cognitive information, e.g., memory or prior knowledge concerning the properties of musical events [Bregman, 1990, pp. 372]. The organization that emerges from this top-down flow of information was coined the term *schema-based organization* by Bregman [Bregman, 1990, pp. 38].

In this way, comprehension is viewed as "a complex interactive process between the straight analysis of sensorial data and the use of hypotheses and expectations derived from previous knowledge" [Bello, 2003, pp. 17]. In fact, the described grouping cues, as well as higher-level cognitive information, are usually exploited in music-listening. Nevertheless, the functions served by memory and experience are the most difficult to simulate.

Music-listening machines, and particularly melody detection systems, could certainly improve their accuracy by taking advantage of previous knowledge about the musical signal (e.g., musical key or instruments present), context information (e.g., recognition of repetitive patterns that could initiate listening predictions) or musicological principles (e.g., voice-leading rules, the set of notes usually employed in a particular key or note transition probabilities), besides the strict analysis of the acoustic signal.

Progressing in this direction, David Temperley proposed an extensive rule-based system for modeling the cognition of basic musical structures [Temperley, 2001] (cited in [Klapuri, 2004, pp. 6]) and David Huron developed an exhaustive study on the derivation of voice-leading rules from perceptual principles [Huron, 2001].

Some authors adopt this mixed bottom-up and top-down architecture, for example in automatic music transcription [Kashino *et al.*, 1995, Martin, 1996, Bello, 2003]. Daniel Ellis approaches the problem of computational auditory scene analysis under a prediction-driven framework that sees analysis as a "process of reconciliation between observed acoustic features and the predictions of an internal model of the sound-producing entities in the environment" [Ellis, 1996, pp. 3].

As will be seen later on, we also recur, up to some extent, to higher-level information, namely in the implementation of the melodic smoothness principle.

H. Competition and Cooperation

The above rules of thumb cooperate and compete in the creation of perceptual objects [Bregman, 1990, pp. 394]. Such competition occurs as well between sequential and

simultaneous integration [Bregman, 1990, pp. 29].

One interesting outcome of cue competition is portrayed in Diana Deutsch's "scale illusion" (cited in [Bregman, 1990, pp. 76]): if a sequence of notes goes along a descending scale and another one is ascending in such a way that their frequency ranges overlap, listeners perceive the upper notes as one part and the lower ones as another. This suggests that frequency proximity is more important than continuity in this situation. On the other hand, if timbre is reinforced, one scale is heard as going down, whereas the other is perceived as going up [Bregman, 1990, pp. 94].

Previous knowledge and expectations also exert influence on the creation of mental constructs. In the previous example, if we knew beforehand what the two possibilities were, we could, by selective attention, focus on each of the two hypotheses regardless of the apparent primitive preference for frequency proximity over continuity.

In terms of cooperation, the tendency to combine harmonically-related stimuli is all the more stressed if, for example, the onsets of the individual sonic elements are very close in time. Therefore, these cues cooperate to form a single perceptual unit.

I. Melody Perception Issues

To conclude, the above rules of thumb, as well as the mechanisms of schema-based organization, apply to the overall auditory organization process and, necessarily, to melody perception.

Identifying and following a melodic stream in a mixture of several different sounds is a problem of sound source separation. Source separation (or segregation) is the process of detecting and organizing a mixture of simultaneous sounds deriving from different physical sources according to the multiplicity of their origins. We, as human beings, are not able to carry out this operation all at once, but instead, we can focus our attention on a meaningful subset at each time. This is the case of melody tracking, where, despite the multiplicity of audio streams in a song (e.g., vocals, guitar, bass or percussion), we can easily follow the main melody.

The emergence of such streams, and particularly the main melodic stream, is a result of both primitive and schema-based mechanisms in the human brain. This entails aspects such as time-frequency proximity, timbre similarity or even intensity similarity [Handel, 1989, pp. 215]. For example, melodic coherence is improved by imposing small pitch intervals between consecutive musical notes, which make segregation unlikely.

An important issue involved in the perception of the main melodic stream in an ensemble is the phenomenon of figure-ground organization in audio. This is related to the "tendency to perceive part of [...] the auditory scene as "tightly" organized objects or events (the figure) standing out against a diffuse, poorly organized background (the ground)" [Handel, 1989, pp. 551].

In this respect, Leonard Meyer suggests that "the musical field can be perceived as containing: (1) a single figure without any ground at all, as, for instance, in a piece for solo flute; (2) several figures without any ground, as in a polyphonic composition in which the several parts are clearly segregated and are equally, or almost equally, well shaped; (3) one or sometimes more than one figure accompanied by a ground, as in a typical homophonic texture of the eighteenth or nineteenth centuries; (4) a ground alone, as in the introduction to a musical work - a song, for instance - where the melody or figure is obviously still to come; or (5) a superimposition of small motives which are similar but not exactly alike and which have little real independence of motion, as in so-called heterophonic textures" [Meyer, 1956] (cited in [Tsur, 2000]). As will be seen in the next section, we are interested in the analysis of songs where a single figure dominates and is accompanied by background pitched and/or percussive instruments.

Several aspects have an effect on the perception of the main melodic stream in ensembles. Namely, Robert Francès, while studying music perception in general, investigated the figure-ground relationship in music [Francès, 1958] (cited in [Uitdenbogerd, 2002, pp. 15]). Basically, the main conclusion of his studies was that a musical part is perceived as figure if its pitch is higher than the accompanying parts.

However, this rule fails in some instances. For example, if the upper notes are more or less constant and the lower ones form a more interesting pattern, the lower notes will more easily catch a listener's attention and, thus, will be heard as figure. Alexandra Uitdenbogerd studied the problem of melody extraction in ensembles, having developed a set of algorithms founded on this idea (commonly referred to as "skyline" algorithms) [Uitdenbogerd, 2002]. Her research was carried out in the symbolic domain.

Besides pitch level, other factors act on the perception of a part as figure, namely frequency proximity and intensity [Uitdenbogerd, 2002, pp. 15]. In this respect, Cristian Francu and Craig Nevill-Manning determined the melody in polyphonic MIDI channels by extracting the notes with higher energy in each time interval, calculated as a combination of the amplitude and frequency of the note [Francu and Nevill-Manning, 2000].

In any case, the music listening and perception is not a static process. As a matter of fact, a listener can consciously shift his attention between different parts of the music ensemble based on with his cultural context, prior experience or interest. This is particularly prominent and well achieved by skilled musicians [Francès, 1958] (cited in [Uitdenbogerd, 2002, pp. 15]).

In our work, we recur essentially to intensity and frequency proximity issues, as will be described in Chapter 5.

2.3. Melody Definition

Melody plays a very important role in music content analysis. Indeed, as Eleanor Selfridge-Field points out, despite its close relationship to other dimensions such as rhythm and harmony, "it is melody that enables us to distinguish one work from another. It is melody that human beings are innately able to reproduce by singing, humming and whistling. It is melody that makes music memorable: we are likely to recall a tune long after we have forgotten its text" [Selfridge-Field, 1998, pp. 4]. Moreover, human beings are able to identify melody-corrupted songs, e.g., when notes are missing or are replaced by incorrect ones [Kim *et al.,* 2000], at least up to a certain limit.

Defining melody in clear and unequivocal terms is not trivial. In effect, the sole concept of melody entails a certain subjectivity: for the same song, different people can have different perceptions of what the main melody is. Furthermore, the concept of melody encompasses various aspects [Kim *et al.,* 2000]: melodies can be monophonic, homophonic, contrapuntal; pitched or purely rhythmic; tonal or atonal; and so forth.

Consequently, an authoritative and global definition of melody is difficult to come up with. Instead, several partial descriptions, comprising different facets of melodic characterization, are often proposed. Some of them are more abstract, others rely on perceptual issues and still others are musicologically-inspired.

The notion of melody is often associated with a *sequence of pitched sounds* [Gómez, 2002]. Namely, in Basic Music[27] melody is defined as "a succession of musical tones".

Building on the idea of sequence of pitches, in Grove Music Online[28] melody is presented as "pitched sounds arranged in musical time in accordance with given cultural conventions and constraints" (cited in [Gómez, 2002], pp. 13). In this somewhat vague definition, a few *subjective issues* involved in the characterization of melody are unveiled, specifically, the "cultural conventions and constraints".

Other descriptions take into consideration related *perceptual and emotional features*. For example, in Wordsmyth[29] melody is presented as "musical sounds in a pleasant order and arrangement". Likewise, Adriano Brandão defines melody as "that aspect of music which ties us to certain songs. [...] We, as listeners, know exactly when a melody pleases us or not. It is something unconscious. [...] Among the musical elements, melody is the one that touches us deepest and the one that is more tightly connected to our innermost feelings." [Brandão, 2004]. In the previous Selfridge-Field's characterization, melody perception is also intimately related to the ways in which people remember music.

Additionally, some definitions clarify the close relationship between melody and

[27] http://www.basicmusic.net/glossary.php

[28] http://www.grovemusic.com/

[29] http://www.wordsmyth.net/

rhythm. Namely, Frank Dorritie describes melody as "a succession of rhythms and pitches" [Dorritie, 2000, pp. 68]. In effect, the concept of rhythm is strongly related to the identification of melodies. Actually, "when identifying a melody, the listener perceives not only the pitch/interval information in the melody, but how those notes correspond to particular moments in time (i.e., rhythm). Rhythm is one dimension in which melodies in general can not be transformed [and still be recognized]." [Kim *et al.*, 2000].

Other definitions take advantage of the relation between melody and rhythm to further exploit the *musicological characteristics* of melody. Wordsmyth provides another description where melody is "a sequence of single tones organized rhythmically into a distinct musical phrase or theme" and "a short musical composition containing one or a few musical ideas; song or tune". Additionally, in the American Heritage Dictionary[30], melody is "a rhythmically organized sequence of single tones so related to one another as to make up a particular phrase or idea". In this characterization, the notions of rhythm, musical idea, phrase and theme are added to the basic concept of sequence of tones. Going a bit further in the direction of musical notation, Dorritie also defines melody as "a horizontal musical line of notation on the staff" and, by analogy to written text, "melody is to a musical work what a paragraph is to a composition" [Dorritie, 2000, pp. 68].

Daniel Levitin suggests a quite broad definition of melody: he describes it as "an auditory object that maintains its identity under certain transformations [...] along the six dimensions of pitch, tempo, timbre, loudness, spatial location and reverberant environment; sometimes with changes in rhythm; but rarely with changes in contour" [Levitin, 1999] (cited in [Kim *et al.*, 2000]). Basically, what this definition says is that melody is a robust feature, which can be recognized after transformations such as intensity or tempo changes, after transpositions, in performances by different instruments or players, in different styles and ornamentations or even when notes are corrupted or missing. In short, its robust identification depends on the stability of its contour.

Given the subjective characterization of melody and the observed diversity in the previous definitions (addressing perceptual, emotional, cultural and musicological facets of melody), one question should now be answered: how do we define melody in the context of our work? Naturally, we do not aim to contemplate all the aspects involved in the notion of melody. In order to narrow the scope of our research, we resort to Selfridge-Field and Dorritie's above definitions. Hence, for the purposes of this work we define melody (or better said, main melody) in this manner:

"*Melody is the dominant individual pitched line in a musical ensemble.*"

In this definition, a few core ideas are summarized, some of them pertaining to the

[30] http://www.bartleby.com/61/

figure-ground organization discussed before.

First, the explicit use of the term *ensemble* unveils our goal of emphasizing the analysis of polyphonic music, i.e., where several instruments are playing concurrently, rather than monophonic music, e.g., folk songs.

Second, streams corresponding to percussion instruments are rejected, in conformity with the *pitched* requirement. Therefore, we exclude from our study styles based solely on percussion, such as purely rhythmic music.

Third, we only consider the class of music where there is a *dominant* voice. In reality, since we are considering musical ensembles, it is necessary to determine what a listener perceives as main melody and what remains as accompaniment. In our case we assume that the melody is conveyed by a dominant voice, where by dominant we mean the "figure", i.e., the part that usually stands out in a mixture due to its pitch level, intensity, contour pattern, etc. This is what the average listener pays attention to more intuitively, e.g., lead vocals in pop music, lead saxophone in jazz or the soloist in opera. In fact, these are the elements that "human beings are innately able to reproduce by singing, humming and whistling", i.e., this is the melody according to Selfridge-Field. Thus, we discard pieces where counterpoint is used (e.g., choral pieces where several simultaneous parts compete against each other without an evident supremacy of any of them).

Finally, through the *individual line* issue, we exclude from the melody the accompaniment parts that stand out when the lead voice is absent. In effect, in polyphonic songs several melodic lines are present (solo, guitar, keyboards, drums, etc.). This in turn directs us to the necessity of separating the main from the accompaniment voices, although it can be argued that, in the perception of melody, the most prominent accompaniment moves to the foreground when the solo is absent. Here, we define *line* as a sequence of notes, each of them characterized by its pitch, starting time, duration, intensity and performance dynamics. Additionally, we assume that the instrument carrying the melody is not changed during the piece (although our algorithm could cope with this situation, as will be clarified in Chapter 5).

Furthermore, we limit our scope to Western tonal music in any of its genres, e.g., pop, rock, classical, jazz, latin, etc. Yet, we could make use of music from other cultures and proveniences, as well as atonal music, provided that our basic requirement is fulfilled: the presence of a principal voice in a polyphonic audio context. On the contrary, music that is not melody-oriented, nor even note-oriented, is out of the scope of our study, e.g., electro-acoustic music.

Without the previous constraints, several specific cases would make the automatic extraction of melody even more complicated, besides the inherent signal processing difficulties. Some of these are listed below, based on [Nettheim, 1992] (cited in [Gómez *et al.*, 2003]):

- A single line played by a single instrument or voice may be formed by move-

ment between two or more melodic or accompaniment strands;

- Two or more contrapuntal lines may equally well claim to bear the melody;
- The melodic line may move from one voice to another, possibly with overlap;
- There may be passages of figuration not properly considered as melody.

The conditions above are not considered in this study.

2.4. Melody Detection in MIR Research

In Section 1.1, we discussed the relevance of melody detection to MIR research and enumerated several related applications, e.g., query-by-melody, automatic music transcription (and melody transcription in particular), plagiarism detection, music education and training, etc.

In this section, we present an overview of music transcription in its main variants, namely, monophonic, melodic[31] and polyphonic transcription. Pitch detection, one of the most important tasks towards this end, is briefly introduced in this section and will be further developed in Chapter 3.

2.4.1. Automatic Music Transcription

Music transcription is a process that aims to convert an arbitrary musical audio signal, e.g., a musical recording such as an mp3 file, into a musical score, identifying pitch, timings, dynamic information and other features.

Besides its possibilities for music composition or education and training, music transcription is also particularly important for MIR research since it establishes the bridge between audio and symbolic MIR [Tzanetakis, 2002, pp. 15]. Indeed, typical research topics of symbolic MIR (theme extraction, motivic analysis, etc.) can then be carried out in the audio realm, once reliable transcriptions are available.

Well-trained people can perform it, at least to some extent. However, several iterations, an educated ear, experience and musical knowledge (in terms of the particular musical style, the instruments involved and their different playing techniques, the harmonic and rhythmic contexts, etc.) are necessary. The same audio file must usually be listened to several times, in order for the transcriber to pay attention to different aspects

[31] Monophonic transcription is sometimes termed "melody transcription", since the only stream present corresponds to the melody. However, in this work, we associate this term with the task of transcribing the main melody in polyphonic recordings.

of the musical piece in each pass.

This can be more or less complex, according to the richness of the polyphony, measured by the type and number of instruments in the mixture. For example, a monophonic recording may require only one iteration whereas transcribing an entire symphony or an ensemble with instruments with similar timbres may turn out to be impossible [Gerhard, 1998, pp. 2]. Particularly, tonally fused sounds, which form a unified musical percept intentionally induced in some musical orchestrations, may be difficult to reverse-engineer, i.e., to separate into the corresponding individual musical notes, even for trained musicians. In effect, it is argued that trying to explicitly unveil the musical notes that are "hidden" in a chimeric sound is perceptually unnatural. In this sense, the mechanism of human music transcription must draw from a conscious mental effort, which, needless to say, demands substantial training and musical proficiency.

Automatic music transcription systems are then proposed as a means to overcome the referred difficulties. The architecture of such systems normally comprises three main stages [Gerhard, 2000, pp. 16]:

i) a frequency analysis step, in which pitch features are extracted from the original musical signal, typically on a frame basis;

ii) a pitch detection stage, in which the fundamental frequencies present in each frame are determined;

iii) and a score generation phase, where a final transcription of the signal is yielded.

A. Monophonic Transcription

Monophonic music transcription is a subset of general polyphonic music transcription, where a single melodic line is played on a single instrument. This is often regarded as a "solved" problem, since monophonic pitch detection is a well-studied subject (see Chapter 3) and full source separation, i.e., separation of all the instruments present in the piece, is not necessary.

Even so, with the purpose of fulfilling strict performance requirements, some specific issues still deserve attention, as for example the transcription of the singing voice or the accurate segmentation of pitch tracks into notes [Chai, 2001, pp. 47; Klapuri, 2004, pp. 3]. In fact, "tracking the pitch of a monophonic music passage is practically a solved problem but *quantization* of the continuous track of pitch estimates into note symbols with discrete pitch and timing has turned out to be a very difficult problem for some target signals, particularly for singing" [Klapuri, 2004, pp. 3]. Namely, the accurate identification of musical notes from sequences of pitches is frequently hard to accomplish, as a result of stylistic performance aspects such as vibrato, tremolo, glissando or legato. Thus, the accuracy in automatic transcription of the singing voice, even for single-voice polyphonies, is behind the one achieved by humans.

In addition, automatic score generation may be complex, even in a monophonic case. In reality, key and time signatures, measure boundaries, accidentals and dynamics are usually difficult to automatically determine [Gerhard, 1998, pp. 15]. Even for humans, this is often not trivial: considerable experience in music analysis and sensibility for particular cases is necessary.

Therefore, many of the existing approaches do not perform transcription in a strict musical sense. Instead, the objective is habitually defined as the identification of the notes present in a given piece, e.g., for obtaining a MIDI representation. Nevertheless, the term *transcription*, as used in the literature on monophonic, polyphonic or melodic transcription, refers indistinctly to systems that explicitly generate scores, systems that output symbolic representations such as MIDI or even systems which only output sequences of pitches without explicit definition of note boundaries. In our work, transcription is conducted up to the identification of musical notes, the reason why we prefer to use the *melody detection* denomination instead of transcription.

The first complete monophonic music transcription mechanism we are aware of is the one devised by Martin Piszczalski and Bernard Galler [Piszczalski and Galler, 1977] (cited in [Gerhard, 2000], pp. 6). This work focused on instruments with a relative strong first harmonic (e.g., flutes), playing at a consistent tempo. The method operated on an STFT front-end, formulated note hypotheses based on amplitude information and identified beam groups, measures, etc., for score generation. After this first work others were proposed, e.g., [Askenfelt, 1979] (cited in [McNab *et al.*, 1996b]), where a method for automatic transcription of folk songs was described. However, it required significant human intervention to correct erroneously transcribed pitches and rhythms.

B. Polyphonic Transcription

Performing pitch detection in a polyphonic context is a much more demanding task. Here, several instruments are usually playing at the same time with strong spectral interference between each other. In this way, problems such as *spectral collisions*[32] and *peak masking* are common, placing additional obstacles to pitch detection, both in terms of the actual extraction of the pitches present and the determination of precise frequency and intensity values. Even for a simple situation such as a guitar chord, it is sometimes hard to detect, in an arbitrary context, all the played notes, without false positives.

The difficulties associated with score generation are also emphasized here. Namely, note determination is more complicated, as a consequence of the referred pitch detection intricacies: if pitches are missing or their frequencies and intensities are not suffi-

[32] The expression "spectral collision" is used to refer to the situation where harmonic components of different concurrent sounds coincide in frequency, i.e., *collide*. In Western tonal music, this is more a rule than an exception.

ciently accurate, detection of note boundaries is not so obvious. Moreover, since instrument separation (and maybe identification) is required, the complexity level increases substantially.

Owing to the described difficulties, most of the existing approaches narrow the scope of analysis. Namely, constraints are typically imposed on the maximum polyphony, on the conditions for allowing simultaneous instruments (e.g., no harmonic collisions tolerated) and on the type of signals to analyze (e.g., artificially constructed signals such as random mixtures of MIDI notes, rather than "real-world" signals with realistic dynamics, isolated instruments, absence of percussion, etc.). Semi-automatic methodologies, where human intervention is required, are also suggested.

Earlier polyphonic transcription mechanisms (e.g., [Moorer, 1977; Chafe *et al.*, 1982; Chafe and Jaffe, 1986; Maher, 1989; Katayose and Inokuchi, 1989; Hawley, 1993]) were very limited regarding the maximum permitted number of simultaneous sounds (the polyphony was often restricted to two voices), as well as the pitch range and the relationships between concurrent sounds. Higher polyphonies were tackled at the expense of limiting the analysis to one single well-studied instrument or by relaxing performance requirements in the output.

Only recently systems were developed, which, despite imposing still many restrictions, could work with polyphonies higher than two notes without being confined to one isolated instrument and attaining reasonable accuracy under the assumed conditions [Kashino *et al.*, 1995; Martin, 1996; Sterian, 1999; de Cheveigné and Kawahara, 1999; Martins, 2001; Tolonen and Karjalainen, 2000; Bello, 2003; Klapuri, 2003; Ryynänen and Klapuri, 2005a]. However, most of the proposed methods are especially concerned with the detection of the correct pitches and not so much with their separation into the respective sources.

In effect, musical source separation is far from being solved, despite the current motivating attempts. In this respect, source separation is only accomplished under specific constraints (e.g., the use of a few previously known instruments), rather than in a general framework. Namely, Kunio Kashino and colleagues conducted some efforts towards the identification of the source of each note with recourse to timbre models, based on pre-stored instrument tone memories [Kashino *et al.*, 1995]. An additional drawback of a few transcription tools is that they only undertake pitch detection in mixtures of isolated tones, and hence note boundary detection is not addressed. In general, these assumedly classify themselves as polyphonic pitch detectors rather than automatic transcription systems.

A review of some attempts towards polyphonic transcription, with particular attention to the polyphonic pitch detection process, is provided in Chapter 3.

A complementary problem to automatic transcription is the analysis of performed music. In this respect, Eric Scheirer implemented an algorithm where the computer,

based on an audio recording and on the respective musical score, identifies differences between the written score and the accomplished performance [Scheirer, 1995]. Such differences may stem from aspects of expression (e.g., vibrato) or from execution errors (e.g., incorrect or missing notes). In his work, the computer acts like a novice: it can listen to the piece and follow the score, but doest not have the required transcription skills yet [Gerhard, 1998, pp. 2].

C. Melody Transcription

As for melody transcription, the main topic of our work, this can be regarded as a problem in between the two previous ones.

Indeed, with respect to pitch detection, we are solely interested in deriving the sequence of F0s (or notes) that convey the main melodic line. This is not as complex as extracting the whole set of F0s in the mixture, but it is still difficult due to the polyphonic context of analysis. Namely, the abovementioned peak masking and spectral collision problems are also present here, with the same consequences in terms of accurate detection of note boundaries. In addition, despite the fact of being only necessary to extract the sequence of pitches corresponding to the melody, this leads in practice to multiple-pitch extraction. In reality, the melodic pitches are not always the most salient ones, as will be discussed in Chapter 3.

Moreover, the separation of the melodic pitches/notes from the accompaniment should be carried out, which has turn out to be difficult. In theory, this is also a subproblem of polyphonic transcription, since there the individual sound sources should be separated. However, this is not yet achievable in a general framework, and so we can affirm that this task is specific to melody transcription.

An overview of the state of the art in this fresh research topic is given in the next paragraphs.

2.4.2. Overview of Research on Melody Detection

Only little work has been dedicated to melody detection in "real-world" songs. Nonetheless, this is becoming a very active area in music information retrieval, confirmed by the amount of work devoted to the MIREX'2004 and 2005 evaluations.

Most existing systems, including ours, are generally founded on a front-end for frequency analysis (e.g., Fourier Transform, autocorrelation, auditory models, multi-rate filterbanks or Bayesian frameworks), peak picking and tracking (in the magnitude spectrum, in a summary autocorrelation function or in a pitch probability density function) and post-processing for melody identification (mostly rule-based methodologies taking advantage of perceptual rules of sound organization, musicological principles, path find-

ing in networks of notes, etc.). One exception is [Poliner and Ellis, 2005a], where the authors follow a different strategy by approaching melody detection as a classification problem using Support Vector Machines. Additionally, in most systems, musical notes are not explicitly determined. Instead, melody extraction is often looked upon as a predominant-pitch detection task, where the result is a predominant pitch line. However, even though the outcome of most algorithms is not a "transcript" in a strict sense, the "melody transcription" denomination is often used, despite being somewhat misleading.

The first work we are aware of is Masataka Goto's Predominant-F0 Estimator (PreFEst) [Goto, 2000; Goto, 2001], where a probabilistic model for the detection of melody and bass lines is devised. The central idea is to model the short-time spectrum of a musical signal as a weighted mixture of adaptive tone models (these were static in the first version of the method [Goto, 2000]). The sound wave is first band-pass filtered, since it is assumed that the melody line has the most significant harmonic structure in middle and high frequency regions. Then, tone models, each defined with a number of harmonics modeled as Gaussian distributions and centered at integer multiples of the corresponding F0 in the spectrum, as well as their weights, are iteratively updated through the expectation-maximization algorithm. Since the weights of the tone models represent the relative prominence of every possible harmonic structure, these weights are interpreted as the F0's probability density function (PDF). Salient peaks in the F0's PDF in each frame are then selected as F0 candidates and tracked in a multiple-agent architecture. The final F0 output will then correspond to the frequencies of the most prominent agent, on the basis of specific salience and reliability measures.

One of the shortcomings of Goto's work is that melody/accompaniment discrimination is not performed. In fact, by selecting the most likely F0 candidate in each frame, pitches from the accompaniment are output even when the melody is absent. Matija Marolt extended Goto's work, aiming to cope with this limitation [Marolt, 2004]. Namely, he adopted the probabilistic pitch estimator proposed by Goto, after which Gaussian Mixture Models (GMMs) were used for clustering the different melodic lines according to their sources. To this end, features such as loudness, pitch stability or onset steepness were extracted. However, as reported by the author, the accuracy of the clustering procedure varied considerably across different excerpts.

This approach was further improved in [Marolt, 2005], with some extensions and simplifications to the previous work. Particularly, melodic seeds, i.e., fragments with well-defined melody, are identified before clustering based on their loudness. Then, the similarity between all melodic seeds is calculated with recourse to pitch, loudness and timbre features, and the computed seed matrix is used as a basis for clustering. *K*-means clustering is performed on the seed similarity matrix, with the possibility of assigning the same seed to more than one cluster. Moreover, cluster merging is also implemented. Melodic lines are then grown from the melodic seeds by adjoining neighboring fragments, in a directed acyclic graph framework. Finally, the melody, i.e., the dominant cluster, is

searched for, according to criteria such as fragment loudness, coverage of melody over time and cluster consistency.

Jana Eggink and Guy Brown suggest a methodology for extracting the melody line played by a solo instrument in a mixture [Eggink and Brown, 2004]. First, F0 candidates are identified using an STFT-based front-end. Pitch tracks are then formed and the main melodic path is looked for in a network comprising all possible candidates over time. This is supported by various local and temporal knowledge sources and subject to some constraints (e.g., a tone must be used from its beginning, etc.). The employed knowledge sources enclose features such as F0 strength, instrument likelihood, relative tone usage or interval likelihood, which are weighted according to their relevance. Instrument recognition receives particular attention here, and the likelihood that a particular tone corresponds to the solo instrument is estimated in each frame. A drawback of this mechanism is that it requires the solo instrument to be known beforehand. In addition, frame-based solo instrument recognition, being an important component of the system, did not perform as accurately as needed.

Paul Brossier makes use of a phase vocoder and harmonic comb matching for predominant-F0 extraction [Brossier, 2004]. In his algorithm, the signal is first pre-processed to enhance medium frequencies, while attenuating the lower and higher spectral regions. This is carried out with recourse to an ARMA A-weighting filter. Next, a phase vocoder is applied and the derived magnitude spectrum in each frame is low-pass filtered and normalized, in order that spurious peaks are smoothed out. Local maxima in the resulting spectrum are then detected and matched against a harmonic comb filter. In this way, the most likely F0 in each frame is identified, based on the number of matched peaks and the energy measured in each of the harmonics. After that, pitch tracking is performed and the obtained trajectory is post-processed by way of median filtering. Heuristic rules are also used, so as to restrict the pitch contour to a more continuous path. Finally, pitch candidates in silence regions, determined by a silence threshold, are discarded. No explicit melody/accompaniment separation is conducted in this system.

Graham Poliner and Daniel Ellis [Poliner and Ellis, 2005a] attend to the melody detection problem as a classification task. Basically, a Support Vector Machine, trained on real multi-instrument recordings as well as synthesized MIDI audio, classifies each frame into one of the equal temperament frequencies (ETFs). To this end, the input acoustic vectors (acquired from the normalized STFT coefficients in each time frame) are mapped to the corresponding, previously labeled, target frequencies. Then, melodic/non-melodic discrimination is performed by energy thresholding: each frame is normalized by the median energy value of the song under analysis and non-melodic segments are discarded based on a global threshold. One attractive facet of this method is that no assumptions about spectral structure are undertaken, contrasting to most of the traditional approaches, which rely on particular frequency structures.

Other systems were presented in the 2004 melody extraction evaluation, organized

as part of the ISMIR'2004 Audio Description Contest (also designated as MIREX'2004 in this document) [MIREX, 2004]. Details on its melody extraction track, concerning particularly the participating algorithms and the evaluation schemes, are provided in [Gómez *et al.*, 2006]. Two of the approaches were not published elsewhere and are briefly described in the following paragraphs.

Namely, Sven Tappert and Jan-Mark Batke developed a mechanism largely inspired on Goto's PreFEst, with a few adaptations especially in the tracking of agents: these contain now four time frames of F0 probability vectors (two of the past, the actual and the upcoming frame). Then, the determination of the path of the principal F0 resorts to the local maxima of the F0's PDF in the four frames. As in the original Goto's method, this algorithm does not discriminate between melody and accompaniment.

Juan Bello assumes that the melody is spectrally located in mid/high frequency regions. Thus, the signal is first pre-processed via high-pass filtering, limiting in this manner the analysis to the regions where the melody is more likely to be present. Next, peaks in the autocorrelation function (ACF) of each time frame are identified and tracked. The resulting melodic and non-melodic fragments are then discriminated with recourse to a rule-based strategy: the melodic path is the one that maximizes the energy while minimizing steep changes in the tonal sequence.

A few other methodologies were published as unreviewed online proceedings of MIREX'2005 [MIREX, 2005].

There, Emmanuel Vincent and Mark Plumbey attack the problem of predominant-pitch extraction under a Bayesian framework, based on a family of probabilistic waveform models [Vincent and Plumbey, 2005]. These monophonic models represent the short-term waveform as a sum of harmonic partials, relative to the most important F0 in each frame, plus residual noise. A probabilistic model learned on a training set is used to represent the amplitudes of the partials. With respect to the residual, a psychoacoustically-motivated prior is utilized. Then, for all possible F0 values, the model parameters are estimated by means of a maximum a posteriori criterion. Finally, the F0 posterior probability in each frame is computed and the corresponding maximum is selected as the main F0 in the respective frame. The melody/accompaniment discrimination issue is not addressed.

In Karin Dressler's approach, sinusoidal-tracks are used as the front-end for predominant-F0 extraction [Dressler, 2005]. First, spectral analysis is performed via the STFT, after which eligible spectral peaks are detected in each frame, based on "distinct spectral features". The kernel of the algorithm is the pitch estimation module, where a perceptually-based magnitude weighting is carried out and the harmonic structure of the system is examined. Next, perceptual cues of sound organization, namely pitch frequency and magnitude proximity, are used to connect consecutive pitches, i.e., to create streams. The melodic pitch line is then identified with recourse to a rule-based scheme, where, for

example, tone successions with intervals above the octave are avoided and notes from middle or higher pitch registers are preferred. This method also resorts to the identification of the most active frequency regions. In this way, the weights of the streams belonging to such regions are increased.

Matti Ryynänen and Anssi Klapuri adapted their polyphonic transcription system [Ryynänen and Klapuri, 2005a] to the melody detection task [Ryynänen and Klapuri, 2005b]. They use an auditory model devised by Klapuri as the front-end for pitch detection [Klapuri, 2005]. There, an input signal is passed through a filterbank comprising 72 band-pass filters (BPF), between 60 and 5.2 kHz. Each sub-band signal is compressed, half-wave rectified and filtered, modeling like this the behavior of the hair cells along the basilar membrane. Then, the STFT is computed in each band and the obtained magnitude spectra are summed across all frequency bands. The resulting summary spectrum is analyzed for F0-detection based on a bank of comb filters, in an iterative detection and cancellation framework. After pitch detection, the algorithm recurs to three probabilistic models to detect the melody: a note-event Hidden-Markov Model (HMM), a silence model and a musicological model. The note-event HMM calculates likelihoods for different notes utilizing the F0s detected in each frame. The silence model identifies the regions where no melody notes are sounding. Finally, in the musicological model, the detected F0s are used for musical key estimation, and between-note transitions are decided on in accordance. The note and silence models form then a network whose optimal path, i.e., the melody, is looked for by means of a token-passing algorithm.

Comparative studies of most of the described approaches were undertaken under the MIREX'2004 and 2005 frameworks and will be presented in Chapter 5.

2.5. Overview of the Proposed Melody Detection System

As referred to in Section 1.2, the extraction of the melodic stream is the focus of our approach, regardless of all the other concurrent sound sources. Thus, we do not aim to isolate each of the instrumental lines present, i.e., we do not perform full source separation. Instead, our goal is to separate the melody from "all the rest".

Apart from the restrictions imposed on the existence of a melody as we have defined it (Section 2.3), we derive a general-purpose strategy for melody extraction in polyphonic audio recordings. Our system is mostly based on a bottom-up processing architecture, where physiological and perceptual cues of sound organization are incorporated, hence replicating the human auditory system to some extent. In addition, a few top-down processing elements are employed, where the reverse also applies: higher-level information is utilized, namely by taking advantage of the melodic smoothness principle. Previous knowledge regarding other common practices, e.g., the use of short ornamental notes, is exploited as well.

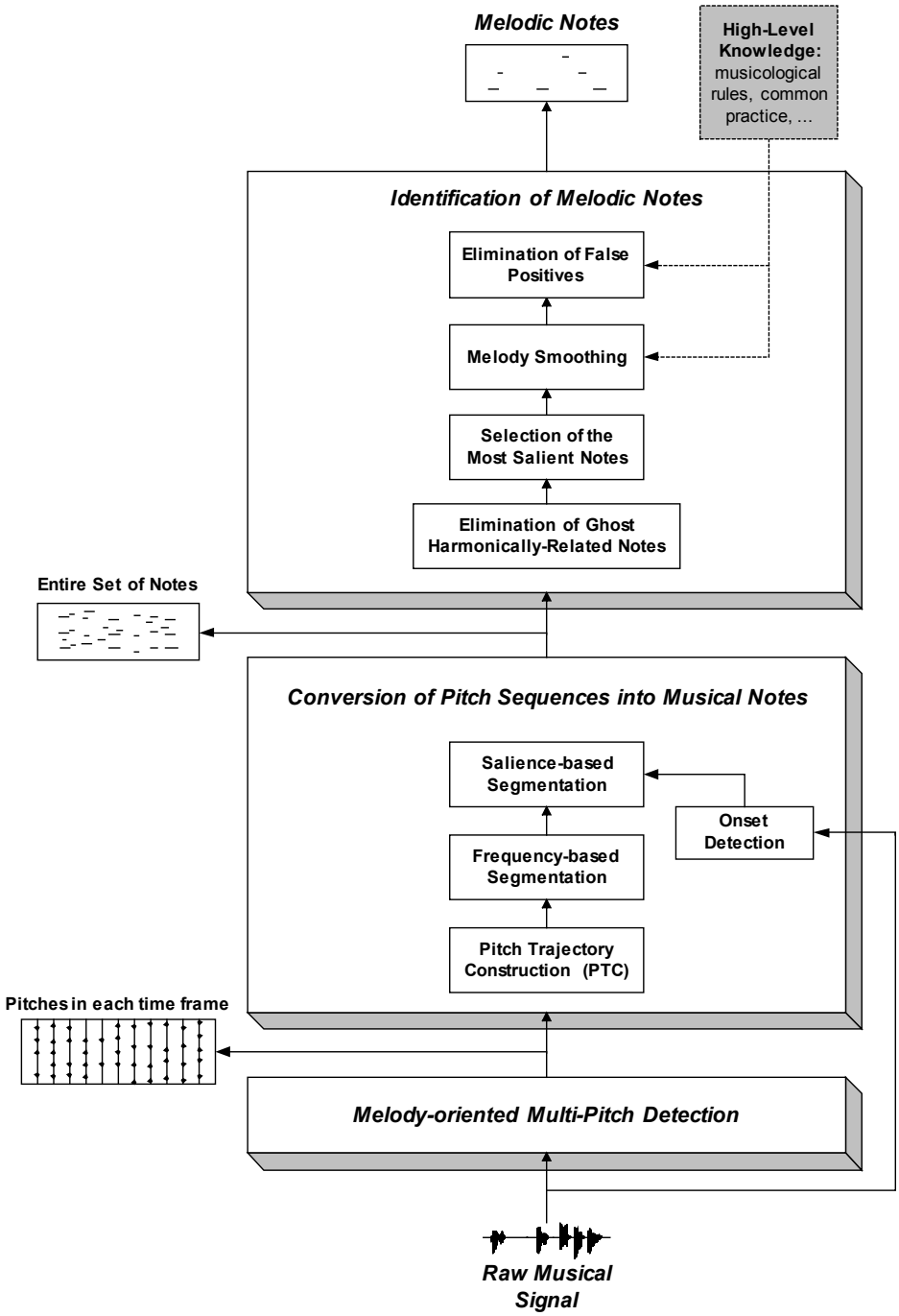


Figure 2.1. Melody detection system overview.

Our approach (sketched in Figure 2.1) entails three main tasks: melody-oriented

multi-pitch extraction, detection of musical notes and identification of the notes conveying the main melodic line. In the figure, the solid arrows indicate that information is flowing bottom-up, whereas the dashed ones denote the reverse case.

The proposed system starts with a raw musical signal, from which low-level pitch features are extracted. In this pitch detection stage, candidate F0s, as well as their saliences, are determined in each frame. These constitute the basis of possible future musical notes. Here, we follow a more pragmatic pitch detection strategy, which seems sufficient for melody detection: instead of looking for all the pitches present in each frame, as happens in general polyphonic pitch detectors, we only capture the ones that most likely carry the melody. These are assumed to be the most salient pitches, which correspond to the highest peaks in a pitch salience curve. Our pitch detection scheme is based on Malcolm Slaney and Richard Lyon's auditory model [Slaney and Lyon, 1993]. Other algorithms were also implemented and compared; however, the auditory-model-based approach proved to perform best.

After pitch extraction, more abstract representations are derived. Namely, musical notes are explicitly identified in terms of pitch, intensity and onset and ending times, maintaining as well the information necessary for the analysis of performance dynamics (e.g., vibrato, glissando) or timbre. This is the goal of the second stage of the method.

To this end, pitch tracks are first created in the Pitch Trajectory Construction (PTC) step, by connecting pitch candidates with similar frequency values in consecutive frames. This is based on the mechanism devised by Xavier Serra [Serra, 1989; Serra, 1997]. The essential idea is to find regions of stable pitches, which indicate the presence of musical notes. In order not to lose information on note dynamics, we took special care to ensure that phenomena such as vibrato or glissando were kept within a single track. Thus, each trajectory may contain more than one note and should, therefore, be segmented in time.

The segmentation of tracks resulting from pitch trajectory construction is performed in two phases: frequency and salience-based segmentation. In frequency-based track segmentation, the goal is to separate all notes of different pitches that might be present in the same trajectory, handling glissando, legato, vibrato and other types of frequency modulation. As for salience-based segmentation, the objective is to separate consecutive notes at the same pitch, which the PTC algorithm may have mistakenly interpreted as forming only one note. This requires trajectory segmentation according to pitch salience minima, which mark the temporal boundaries of each note. To increase the robustness of the method, note onsets are detected directly on the audio signal and used to validate the candidate salience minima found in each pitch track.

In the last stage, we aim to identify the notes that convey the melody among the whole set of obtained notes. We found our strategy on the salience and melodic smoothness principles. Moreover, melody/accompaniment discrimination is attempted.

In this way, ghost harmonically-related notes, i.e., notes whose frequency compo-

nents are sub or super-harmonics of a true note's F0, are first eliminated. Here, we make use of perceptual rules of sound organization, specifically harmonicity and common fate, where common frequency and amplitude modulation are exploited [Bregman, 1990, pp. 227-292].

Although many ghost notes are discarded at this point, a high number of non-melodic notes is still present. Hence, we have to extract the ones that bear the main melody. This is performed in conformity with the salience and melodic smoothness principles. In the salience principle, initial melodic note candidates are identified as corresponding to the most intense notes in the mixture. Besides this purely bottom-up information processing module, a top-down scheme is also used, in which musicological principles of tonal music composition are employed in order to smooth out the initial tentative melody. Namely, it is well known that small pitch intervals are preferred in Western tonal music. Thus, abrupt pitch transitions are examined so as to check their validity, in accordance with the melodic smoothness principle.

Other common practices are explored as well, for example as regards the properties of intensity and duration contours. In reality, sudden intensity or duration variations are not usual in Western music, and so such cases suggest the presence of erroneous notes. Consequently, notes corresponding to fast reductions in intensity or duration are discarded, as they are likely to represent false positives. Short ornamental notes are an exception, and so their possible presence is inspected. Further melody/accompaniment discrimination is carried out by way of note clustering.

The abovementioned higher-level rules are implemented as post-processing stages. These allow for the correction of errors arising from simple low-level feature analysis, e.g., selecting notes solely based on their intensities. In addition, other musicological information is utilized, for example concerning the typical note durations in the Western music canon.

The proposed system was developed in Matlab (version 7)[33], except for a few third-party functions that were coded in the C programming language (see Section 3.6). Performance tests were conducted on a PC with a 3 GHz clock frequency Intel Pentium 4 processor and 512 MB of RAM, running Microsoft Windows XP Professional, version 2002, Service Pack 2.

2.6. Test Collections and Evaluation Procedures

The accurate evaluation of melody extraction systems is difficult to attain for two main reasons: the lack of standard, comprehensive and sizeable databases, as well as quantita-

[33] http://www.mathworks.com/products/matlab/

tive evaluation procedures.

This problem is now attenuated to some extent thanks to the MIREX initiative, which has given a most important contribution to fill in this gap. In 2004 and 2005, two different collections were devised for the comparison of melody extraction systems. Unfortunately, but justified by the difficulties in producing reliable ground truth data, the MIREX'2005 database was not made public to the research community. Therefore, the only entirely accessible standard compilation is the one created as part of MIREX'2004.

2.6.1. Acquisition of Ground Truth Data

Apart from the problems in accessing copyrighted material, one of the most complex issues in the development of test-beds for melody extraction evaluation is the acquisition of reliable and meaningful ground truth annotations. Such annotations must be acquired either manually or automatically.

Manual annotation is typically fulfilled with recourse to visual spectrogram analysis, where note boundaries can be identified to some extent. Naturally, this is not trivial in polyphonic mixtures. Hence, repetitive and localized listening of the song excerpts is normally executed in parallel, to support and improve the temporal accuracy of the observed note boundaries.

This is clearly a time-consuming, error-prone and subjective task. In fact, no established standard rules have been agreed upon as to melody annotation in polyphonic audio recordings[34]. Besides, reliable annotation of the singing voice is complicated by unvoiced components (e.g., fricatives and plosives). In the same way, accurate identification of the temporal boundaries of musical notes may be complex in the presence of strong vibrato and legato. Increased robustness demands specialized skills and concurrent annotations, which, in turn, give rise to substantial man-work. Thus, this approach is highly unpractical and costly. Moreover, in case exact pitches are necessary (rather than F0s quantized to the ETFs), manual annotation fails.

Due to the difficulties of manual annotation, automatic strategies are required. One possibility is to use MIDI synthesized songs, which have the advantage of wide availability. The main drawback comes from the artificiality of synthesized music, which usually simplifies the conducted analysis. In effect, synthesized music does not retain the authentic acoustic complexity of genuine recordings, since important dynamics are not faithfully replicated. Even though such "toy" problems can give good insights on the kinds of techniques to research, "real-world" situations place many more difficulties that might not be satisfactorily dealt with by techniques proposed for artificial problems. As

[34] A contribution towards a general framework for manual annotation of musical audio is described in [Lesaffre *et al.*, 2004].

an example, we developed a mechanism for frequency-based segmentation of pitch tracks (described in Section 4.3), motivated by the necessity to cope with realistic glissando and vibrato, typical in the singing voice and several musical instruments. If only synthesized samples had been used, the artificially-generated dynamics would be too well-behaved, and probably we would not have found the need to handle the encountered difficulties.

For "real-world" songs, robust (semi-)automatic annotations can only be achieved when multi-tracking recordings are available. In this case, well-studied monophonic pitch detection and note segmentation algorithms may be employed on the melody channel. Pitch extraction as well as segmentation errors are, nevertheless, expected, which requires manual inspection and correction. Anyway, the human labor and the required skills are by no means comparable to the ones of manual annotation. Furthermore, exact pitch values, rather than quantized F0s, are obtained. This solution is, however, limited by the availability of copyright-free multi-track recordings, which leads in practice to small-sized databases. In spite of this restriction, this solution represents the best compromise for deriving reliable ground truth data, and was the one followed in both MIREX'2004 and 2005 [MIREX, 2004; MIREX, 2005] (although in the MIREX'2004 set some synthetic samples where utilized anyway).

Before the creation of standard test-beds, it was common that each author compiled his own collection, defining the intended music style, instrumentation and acoustic characteristics, as well as the metrics of evaluation, according to his own criteria. In this way, besides using the MIREX'2004 database, we have also evaluated each module of our melody extraction system with a test-bed we had previously assembled (Table 2.1).

Both databases were designed taking into consideration their diversity and musical content[35]. In reality, for a meaningful evaluation to be accomplished, the musical material should cover a variety of styles. This is necessary in order to evaluate the performance of melody detection systems in a general framework.

In our test-bed (abbreviated as PDB, for personal database), we collected excerpts of about 6 seconds from 11 songs (the topmost ones in Table 2.1), enclosing several different categories. The selected excerpts were manually annotated with the correct notes, in conformity with the abovementioned methodology. Contrariwise to our previous assumptions, we also selected a choral piece (ID 2), consisting of four simultaneous voices plus orchestral accompaniment. The idea was to study the behavior of the algorithm in this situation, where we defined the solo as corresponding to the soprano.

As for the MIREX'2004 database (hereafter designated as M04), we adopted the defined training set. Namely, 2 items of synthesized singing voice plus background music (daisy2/3), 2 items of saxophone melodic phrases with accompaniment (jazz2/3), 2 items

[35] Further details on the used excerpts are given in Appendix B. These, as well as annotation and result files, can be downloaded from http://www.dei.uc.pt/~ruipedro/MelodyDetection/.

consisting of a MIDI synthesized polyphonic sound with a predominant voice (midi1/2), 2 items of opera singing, one with a male and another with a female soloist, plus orchestration, and 2 items of sung pop music plus accompaniment (pop1/4) [Gómez *et al.*, 2006; MIREX, 2004]. The selected excerpts, each of around 20 seconds, were (semi-)automatically annotated, complying with the referred methodology (i.e., monophonic pitch detection in the melodic track, using multi-track recordings). Another 10 similar excerpts were used just for testing purposes (not used for training).

ID	*Song Title*	*Category*	*Solo Type*
1	Pachelbel's "Kanon"	Classical	Instrumental
2	Handel's "Hallelujah"	Choral	Vocal
3	Enya – "Only Time"	New Age	Vocal
4	Dido – "Thank You"	Pop	Vocal
5	Ricky Martin – "Private Emotion"	Pop	Vocal
6	Avril Lavigne – "Complicated"	Pop/Rock	Vocal
7	Claudio Roditi – "Rua Dona Margarida"	Jazz/Easy	Instrumental
8	Mambo Kings – "Bella Maria de Mi Alma"	Bolero	Instrumental
9	Eliades Ochoa – "Chan Chan"	Son	Vocal
10	Juan Luis Guerra – "Palomita Blanca"	Bachata	Vocal
11	Battlefield Band – "Snow on the Hills"	Scottish Folk	Instrumental
12	daisy2	Pop	Vocal
13	daisy3	Pop	Vocal
14	jazz2	Jazz	Instrumental
15	jazz3	Jazz	Instrumental
16	midi1	Pop	Instrumental
17	midi2	Folk	Instrumental
18	opera female 2	Opera	Vocal
19	opera male 3	Opera	Vocal
20	pop1	Pop	Vocal
21	pop4	Pop	Vocal

Table 2.1. Description of used song excerpts. Excerpts 1-11: personal database; excerpts 12-12: MIREX'2004 training set.

Additionally, we evaluated the algorithm on the MIREX'2005 database (25 excerpts of around 10 to 40 seconds). The collected audio files were not made public and so only average results are provided (Section 5.8).

The songs in both databases contain a solo (either vocal or instrumental), which was defined as the target melody, plus accompaniment parts (guitar, bass, percussion, other vocals, etc.). Furthermore, we have identified a few other requirements, particularly while devising our personal test set, as follows:

i) absence of the solo in a few time intervals, in order to evaluate melodic/non-melodic discrimination capabilities of the system;

ii) existence of octave-related notes, necessary to assess the robustness of the method against octave errors (besides the ones that occur due to the selection of harmonic peaks in the pitch detection stage - Chapter 3);

iii) occurrence of intense non-melodic notes and percussion, an important aspect to test as a result of our idea of selecting the most salient notes at each moment; this is related to the signal-to-noise ratio of the piece under study. As previously referred to, we define SNR as the relation between the intensity of the melodic instrument and the intensity of the background.

iv) presence of notes with real dynamics (glissando, legato, vibrato, tremolo and other sorts of frequency and amplitude modulation), as well as consecutive notes at the same pitch, with the purpose of evaluating note determination accuracy; to this end, different types of instruments were utilized, including the singing voice, rather than constraining the algorithm to a particular instrument.

We employed 16-bit Pulse Code Modulation wave files with monaural recordings, sampled at 44.1 kHz (CD quality), except for the ones in our test-bed, where the sampling rate was 22.05 kHz. We defined this sampling rate because it proved sufficient and was computationally more efficient. Anyway, a sampling rate of 44.1 kHz was specified in both MIREX'2004 and 2005.

Although the used material is already quite valuable for evaluation, expressive conclusions can only be drawn if sizeable data sets are assembled. For instance, it is difficult to conduct analysis on style dependencies when only a few jazz or pop excerpts are available. However, many practical difficulties are involved here, as it was described.

2.6.2. Evaluation Metrics

As for evaluation procedures, standard and meaningful quantitative metrics are indispensable as well.

Depending on the application, the melody might be output as a sequence of notes or as a continuous pitch contour. For example, in tasks such as audio-to-midi conversion or query-by-humming, notes should be explicitly determined. In other jobs, e.g., analysis of performance dynamics (vibrato, glissando, etc.), pitch contours are preferred.

In our system, we are particularly interested in extracting musical notes, although pitch track contours are also accessible. Moreover, in our test-bed, we do not know the exact target frequencies and so we measure MIDI note extraction accuracy. Concerning the M04 database, since both exact frequencies and quantized notes are available, the two possibilities are evaluated.

A. Pitch Contour Accuracy

Regarding pitch contour accuracy, the MIREX initiative gave, once again, a crucial impulse with the definition of a number of metrics for evaluation of melody extraction algorithms. These metrics take into account aspects such as raw and chroma pitch accuracy, the occurrence of octave errors or the melodic/non-melodic discrimination ability, in a frame-based analysis [Gómez *et al.*, 2006].

Overall Raw Pitch Accuracy (ORPA)

This metric consists on a frame-wise comparison of the annotated and the extracted pitch contours. Given that musical pitches in the equal temperament tuning are distributed along a logarithmic scale (e.g., [Martins, 2001, pp. 75]), the F0s in Hz units, f_{Hz}, are converted to cents[36], f_{cent}, according to (2.1) [Gómez *et al.*, 2006]. This applies to both the annotated and the extracted F0s.

$$f_{cent} = \begin{cases} 1200 \cdot \left[\log_2 \left(\frac{f_{Hz}}{13.75} \right) - 0.25 \right] & , f_{Hz} \neq 0 \\ 0 & , f_{Hz} = 0 \end{cases} \tag{2.1}$$

Then, the pitch error in each frame is measured by averaging the absolute difference between the annotated pitch value and the extracted one. This error is bounded to a maximum of one semitone, i.e., 100 cents, as in (2.2):

$$err[i] = \begin{cases} 100, & if \quad \left| f_{cent}^{ext}[i] - f_{cent}^{ref}[i] \right| \geq 100 \\ \left| f_{cent}^{ext}[i] - f_{cent}^{ref}[i] \right|, & otherwise \end{cases} \tag{2.2}$$

[36] Briefly, the basic musical interval in the equal temperament scale is the cent. An interval of 100 cents is a semitone and 1200 cents form an octave.

where $f_{cent}^{ext}[i]$ and $f_{cent}^{ref}[i]$ denote, respectively, the extracted and annotated F0s in the i^{th} frame, and *err*[*i*] stands for the absolute pitch detection error in the same frame[37].

As a convention, non-melodic frames are assigned target frequencies of 0 Hz. Therefore, in case of inaccurate melody/accompaniment discrimination, the error will generally be maximum in such frames (i.e., 100).

The final score (on a 0-100 scale) is obtained by subtracting the bounded mean absolute difference from 100, as in (2.3) [Gómez *et al.*, 2006]. There, *N* stands for the total number of frames in the excerpt under analysis.

$$score = \quad 100 - \frac{1}{N} \cdot \sum_{i=1}^{N} err[i] \tag{2.3}$$

In the described evaluation methodology, each frame contributes to the final result with the same weight. In this way, this metric evaluates pitch detection performance using both melodic and non-melodic frames, thus indirectly evaluating the capability of the system to separate the melody from the accompaniment. Hence, systems with very good pitch detection accuracy but insufficient melodic/non-melodic discrimination abilities will be penalized.

Melodic Raw Pitch Accuracy (MRPA)

In order to evaluate pitch detection performance taking into consideration only the melodic frames (i.e., ignoring melodic/non-melodic discrimination), the same score is computed, this time using only those frames. Formally, it comes (2.4):

$$score = \quad 100 - \frac{1}{N_m} \cdot \sum_{i \in \{melodic\ frames\}} err[i] \tag{2.4}$$

where N_m stands for the number of annotated melodic frames.

Melodic Chroma Pitch Accuracy (MCPA)

This metric is similar to the previous one, except that octave errors, a common problem in pitch detection algorithms, are now disregarded.

Here, both the annotated and the extracted F0 values are mapped to the range of one octave before calculating the absolute error, according to (2.5):

$$f_{chroma}^{ext}[i] = 100 + \mathrm{mod}\left(f_{cent}^{ext}[i], 1200\right), \quad i : f_{cent}^{ext}[i] \neq 0 \tag{2.5}$$

[37] In terms of notation, we follow the common practice of using square brackets for the indexes of discrete variables (as in *err*[*i*]) and parentheses for continuous ones.

There, f_{chroma}^{ext} denotes the chroma values of the melodic F0s, i.e., the original values mapped to the range of one octave. The chroma annotated F0 is derived likewise. In (2.5), the 100 offset was set in order to prevent chroma values of 0 cents for multiples of 1200, which would be confused with the target values for non-melodic frames.

The errors and score are then calculated as before, i.e., according to (2.2) and (2.3). However, since the maximum error is now half an octave, error values above 600 cents must be corrected following a "circular reasoning". It comes then (2.6):

$$err_{circular}[i] = \begin{cases} 1200 - err[i], & if\ err[i] > 600 \\ err[i], & otherwise \end{cases} \tag{2.6}$$

where $err_{circular}[i]$ denotes the circular error in frame i.

B. Note Extraction Accuracy

As for note extraction accuracy, metrics based on the percentage of correctly extracted notes, as well as on the edit distance, can be used.

Percentage of Correct Frames

With respect to note accuracy, we could simply count the number of correctly extracted notes and divide it by the total number of annotated ones. But in order to accomplish a more precise figure that could cope with notes with different lengths, duration mismatches, etc., we decided to compute the note accuracy metric as the percentage of correctly identified frames. There, the target and the extracted frequency values in each frame are defined as the ETFs of the corresponding notes. The error and overall score are then calculated in the same described manner.

Thus, we define three metrics, related to the previous ones: *melodic raw note accuracy* (MRNA), *melodic chroma note accuracy* (MCNA) and *overall raw note accuracy* (ORNA).

Since we do not know the exact target frequency values for the excerpts in our testbed, we employ preferably note metrics. Even so, pitch performance figures are also provided for completeness and for comparison with other melody extraction systems that extract pitch contours rather than melodic notes.

Melodic Similarity Metric (MSM)

Besides this approach, another note-based evaluation is proposed in [Gómez *et al.*, 2006; MIREX, 2004]. There, an edit distance between the extracted and the annotated melodies is computed as the cost of transforming one melody into the other. To this end, different penalizations are assigned to the required transformation operations, namely, insertions, deletions and substitutions. In the MIREX'2004 evaluation, the edit

distance was obtained as described in [Grachten *et al.*, 2002].

The main disadvantage of this measure comes from the difficulty in interpreting the resulting numbers in absolute terms. For example, it is difficult to determine the perceptual significance of a distance of 5. Nevertheless, we always know that a distance of 4 is better than a distance of 5. Thus, this metric is more adequate for relative scales, i.e., for comparing the performance of different algorithms regardless of the absolute meaning of the attained values. For this reason, we will not use it in the evaluation of our system but will present the results achieved in the MIREX'2004 evaluation, where it was computed.

Anyway, despite the importance of explicitly identifying the musical notes in a song, this is a somewhat underestimated topic in the field of melody detection [Gómez *et al.*, 2006] (see Chapter 4), attested by the absence of note-oriented metrics in the MIREX'2005 evaluation. In effect, few existing methods address this topic.

C. Other Melody Discrimination Metrics

Finally, we calculate two other statistics in order to evaluate the ability of the system to separate the melody from the accompaniment, namely recall and precision.

Recall

Recall is the percentage of annotated non-melodic frames that the system classifies correctly as non-melodic. Formally, it comes (2.7):

$$recall = \frac{TN}{TN + FP} \tag{2.7}$$

where *TN* (True Negatives) stands for the number of non-melodic frames correctly classified as non-melodic and *FP* (False Positives) denotes the number of non-melodic frames erroneously classified as melodic. Hence, *TN+FP* is the total number of annotated non-melodic frames.

Precision

Precision is the percentage of extracted non-melodic frames that are indeed non-melodic. Formally, it turns out (2.8):

$$precision = \frac{TN}{TN + FN} \tag{2.8}$$

There, *FN* (False Negatives) denotes the number of melodic frames erroneously classified as non-melodic. Thus, *TN+FN* is the total number of frames classified as non-melodic by the system.

Chapter 3

Pitch Detection

> *"Some vibratory impulses or motions causing a percussion on the ear return with greater speed than others. Consequently, they have a greater number of vibrations in a given time, while others are repeated slowly, and consequently are less frequent for a given length of time. The quick returns and greater number of such impulses produce the highest sounds, while the slower, which have fewer vibrations, produce the lower."*

Euclid (365 BC - 275 BC), "Elements of Music, Introduction to the Section of the Canon"

Pitch is the main low-level feature in melody detection. Much work has been devoted to pitch detection throughout the years, mostly in the analysis of monophonic speech signals. More recently, pitch detection methodologies have been devised to deal specifically with musical signals, both in monophonic and polyphonic contexts. Such approaches are discussed in this chapter, with the purpose of defining a pitch detection strategy that might be adequate to our ultimate melody detection goals. Namely, the purpose of the first stage of our system (in Figure 2.1) is to capture the most salient pitch candidates at each time, which constitute the basis of possible future notes.

The fact that our final objective is melody detection would probably suggest that extracting the most prominent pitch at each time would suffice. However, since we are working in a polyphonic context, pitches from the accompaniment might compete for predominance with the ones from the melody, being alternately selected. Furthermore, ghost pitch candidates that are super or sub-harmonics of true pitches might be selected as well, causing octave (or, generally speaking, harmonic) errors. Therefore, it is insufficient to select only one fundamental frequency at each time. Instead, several pitch candidates must be picked up so that the one corresponding to the melody is likely to be present.

Section 3.1. Introduction

We start this chapter with some context information on the topic of pitch detection. Namely, the concepts of harmonic sounds, fundamental frequency and pitch are first introduced. The pitch detection mechanism is then outlined, briefly describing its three main phases: pre-processing, extraction and post-processing. Finally, we review some of the existing strategies for both monophonic and polyphonic pitch extraction.

Section 3.2. Pre-Processing: RASTA Processing

We then address the problem of additive and convolutive noise suppression in musical signals. Here, we describe an algorithm proposed by Anssi Klapuri [Klapuri, 2003], based on the principles of RASTA processing [Hermansky *et al.,* 1993].

Section 3.3. Extraction: Auditory-Model-based Pitch Detector

We have analyzed, implemented and compared different types of pitch detectors. From the conducted study, an auditory-model-based approach (using Slaney and Lyon's auditory model [Slaney and Lyon, 1993]) is chosen due to its improved detection accuracy. This pitch detector is discussed in detail in this section, namely its main components: the ear model, channel periodicity analysis, periodicity summarization (leading to a summary autocorrelation function – SACF) and salient peak detection. The other evaluated pitch detection methods are described in Appendix A.

Section 3.4. Post-Processing: SACF Enhancement

The last phase of pitch detection is described in this section. Here, we employ the algorithm conceived in [Tolonen and Karjalainen, 2000]. The idea is to enhance the SACF, aiming to remove much of the noisy and redundant information in it, namely peaks corresponding to sub or super-harmonics of the fundamental.

Section 3.5. Putting It All Together

The complete pitch detection procedure is summarized in algorithmic form and model parameters are listed in this section.

Section 3.6. Experimental Results, Analysis and Conclusions

Finally, experimental results relating to this module are presented. Moreover, a comparative analysis of the evaluated pitch detectors is conducted. The main advantages and shortcomings of the followed approach are discussed and pointers for future improvements are provided.

3.1. Introduction

Before attending to the problem of pitch detection, it is important to clarify its concept. This is carried out in the next subsections, after which different methods for single and multiple-pitch detection are reviewed.

3.1.1. Harmonic Sounds, Fundamental Frequency and Pitch

Pitch is classically defined as the tonal height of a note, i.e., "that attribute of auditory sensation in terms of which sounds may be ordered on a scale extending from low to high" [ANSI, 1994]. Therefore, a pure tone of 500 Hz has a higher pitch than a pure tone of 400 Hz. This example suggests a connection between pitch and frequency. In fact, pitch is the perceptual correlate of the fundamental frequency of a tone and is often described as "the perceived F0 of a sound".

It is common to distinguish between physical and perceptual properties of musical sounds. Namely, "physical properties of sounds are those that can be measured directly using scientific instruments", whereas "the perceptual attributes of sounds are those that a human listener associates with the sound" [Scheirer, 2000, pp. 53]. Some of these perceptual variables are immediately correlated to physical ones; this is the case of pitch and fundamental frequency.

The concept of fundamental frequency can be best explained with recourse to the idea of harmonic sound. Briefly, a harmonic sound is one that can be decomposed into a sum of sine waves whose frequencies are (approximately) integer multiples of a frequency F0. This frequency is the *fundamental frequency* and the integer multiple frequencies are named *harmonics*. In this way, harmonic sounds are periodic (or "almost" periodic, as described in the following paragraphs).

From the sinusoidal model, it transpires that, besides being periodic, harmonic sounds have a spectral structure where the outstanding frequency components, i.e., F0 and harmonics, are regularly spaced. This is illustrated in Figure 3.1 for a saxophone sound with an F0 of around 370 Hz and a corresponding period of 2.7 msec.

As can be noticed in Figure 3.1a, the signals of real-word harmonic sounds, generated for example by so-called pitched musical instruments (e.g., woodwind, string, reed or brass instruments, the human singing voice, etc.) are not strictly periodic; instead, their cycles are slightly different from each other. Hence, they are denominated *pseudo-periodic* signals. Nevertheless, definite pitches can be assigned to the sounds they produce.

In contrast, non-harmonic sounds such as the ones produced by most percussive instruments (especially in the class of membranophones, e.g., snare drums) do not show

clear periodicity and, thus, are not the subject of F0 detection. These are termed unpitched instruments, since their sounds contain complex frequencies from which no definite pitch can be discerned.

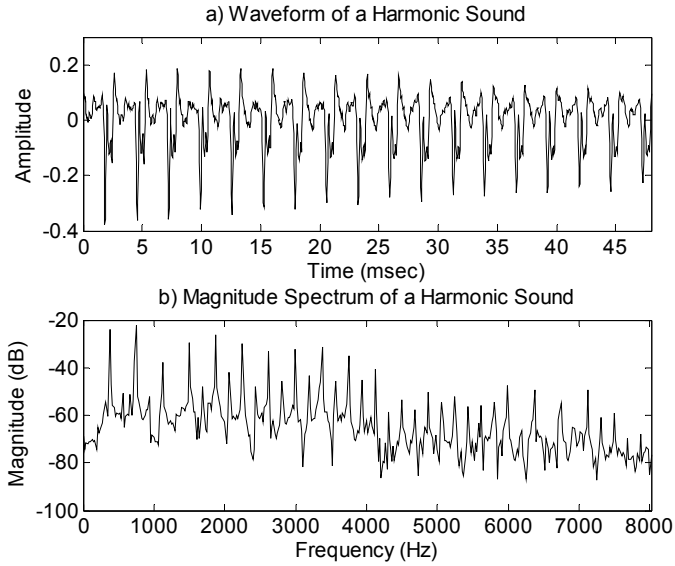


Figure 3.1. Time and frequency-domain illustration of a harmonic sound.

One notable exception concerns sounds without a regular harmonic structure, but that are nearly periodic. This is the case of mallet percussion instruments (e.g., xylophone, marimba), which produce non-harmonic pitched sounds [Klapuri, 2004, pp. 22]. Typically, such instruments do not produce harmonic overtones, although a few partials are close to integer multiples of the F0.

Additionally, in an ideal harmonic sound the frequencies of the harmonics are exact integer multiples of the F0. However, in real-world sounds the harmonics do not perfectly match their theoretical values; instead, they depart somewhat from their ideal frequencies - a phenomenon designated as *inharmonicity*, i.e., non-ideal harmonicity. For instance, in the case of stretched strings, e.g., in pianos, higher-order harmonics are shifted upwards in frequency by an almost constant factor [Klapuri, 2004, pp. 22].

Inharmonicity is one of the factors that give each sound a particular timbre, as will be seen in Section 5.6. Anyway, although perfect harmonicity does not occur in practice, the general structure of musical sound spectra is similar to the one in Figure 3.1.

As mentioned, pitch is the perceptual correlate of fundamental frequency. The perception of a pitch frequency depends on the F0 of the sound, but also on its intensity, as well as on the listener and environment. Perceived pitches increase about one octave

with every doubling in F0, and so pitch frequencies are related to the logarithm of F0 values. This relationship is not strictly logarithmic, since above 1000 Hz F0 doublings lead to the perception of an interval slightly less than an octave [Gerhard, 2000, pp. 16]. Moreover, the relationship between F0 and pitch changes as well with sound intensity and harmonic content. For example, the pitch perception of two signals with equal F0 may be different, for example, in case one of them shows perfect harmonicity whereas in the other the harmonics deviate somewhat from their ideal theoretical values. Indeed, harmonicity concurs to pitch distinctness [Gerhard, 2003, pp. 2].

One of the most interesting examples of the difference between pitch and F0 is the phenomenon of the missing fundamental [Bregman, 1990, pp. 237]. In effect, if, for instance, a tone is created with only the 3rd to 5th harmonics, the F0 that is common to those harmonics is still "heard" despite its absence in the spectrum. This implies that, for pitch perception, the frequency spectrum of the signal is at least as important as the F0.

The centrality of pitch in hearing is attested by the fact that the human auditory system tries to assign a pitch to almost all kind of acoustic signals, either harmonic or not. Besides (pseudo-)periodic signals, noise sounds can sometimes be matched with a sinusoid of a specific frequency. For example, if we take a random noise signal and amplitude modulate it, a pitch frequency is perceived, which corresponds to the modulating frequency. In addition, the auditory system may also assign a pitch to sounds that are neither evidently periodic nor show a regular spectral structure, e.g., bells or vibrating membranes. In reality, the human auditory system seems to have a natural tendency to compact certain aspects of sound events by using a single frequency [Klapuri, 2004, pp. 21].

From the previous description, it turns out that pitch and fundamental frequency are two related yet different concepts. Nevertheless, the two terms are frequently employed as synonyms in the pitch detection literature, although most of the work in this field is actually concerned to F0 extraction. We too use both expressions interchangeably throughout the text. In this way, pitched sounds are assumed to be the ones that have a clear fundamental frequency, as generally happens in harmonic sounds.

3.1.2. The Pitch Detection Process

Based on the previous discussion, the essential problem of pitch detection is then to determine the fundamental frequencies present at each time in an audio signal. The analysis conducted in the next paragraphs follows an overview presented in [Gómez *et al.*, 2003].

Much work has already been devoted to this topic, especially in the monophonic domain. In fact, pitch detection is an important task in both speech and music content analysis. As referred to in Chapter 1, despite some difficulties, e.g., in the processing of

the singing voice and handling of octave errors, monophonic pitch detection is usually regarded as "practically a solved problem" [Klapuri, 2004, pp. 3], with several reliable and real-time algorithms available (e.g., [de Cheveigné and Kawahara, 2002; Doval and Rodet, 1991; Noll, 1967]). More recently, several approaches have been devised for polyphonic pitch detection in musical signals, e.g., [Klapuri, 2004; Sterian, 1999; Kashino *et al.*, 1995].

Wolfgang Hess [Hess, 1983] (cited in [Gómez *et al.*, 2003]) condensed the pitch detection process into three main sequential stages (Figure 3.2): the pre-processor, the extractor and the post-processor. This general architecture was proposed in the context of monophonic pitch detection, but its main building blocks also apply to polyphonic analysis.

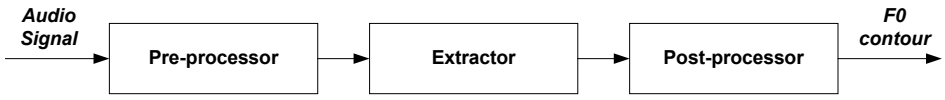


Figure 3.2. Overview of the monophonic pitch detection process.

The objective of the pre-processor is to perform data reduction so as to facilitate the F0 extraction procedure. Namely, one of its main tasks is to suppress noise as a means of improving pitch detection accuracy. Another goal is to enhance the features that are useful for F0 determination.

After pre-processing, the extractor - the core pitch detection module - examines the obtained signal and looks for the fundamental frequencies in each frame. Different strategies for both monophonic and polyphonic pitch detection are described in the next subsections.

Finally, in the post-processor, several tasks such as error detection and correction, smoothing, etc., might be carried out. Indeed, the resulting F0 contour is usually noisy. Namely, it is often affected by isolated errors, e.g., incorrectly extracted outliers or sub/super-harmonics.

3.1.3. Monophonic Pitch Detection

The first attempts towards pitch detection in monophonic musical signals brought in methods from the speech research community. More recently, new techniques have been specifically designed for music. In effect, musical signals have some peculiarities that require particular attention [Klapuri, 2004, pp. 79]. Namely, the pitch range of musical sounds is wider than that of speech signals and the spectral content of the sounds produced by musical instruments vary significantly. Also, phenomena such as inharmonicity

should be taken into account.

Monophonic pitch detection is usually considered solved, since satisfactory results can be achieved in most situations. Nevertheless, strict demands are hard to fulfill in some specific problems, e.g., in the processing of the singing voice or in the accurate analysis of the attack and decay regions of notes.

Despite the myriad of different pitch detection algorithms that have been proposed since the 1960's until our days (e.g., [Noll, 1967; Gold and Rabiner, 1969; Lahat *et al.,* 1987; Slaney and Lyon, 1990; Doval and Rodet, 1991; Maher and Beauchamp, 1993; Talkin, 1995; de Cheveigné and Kawahara, 2002; Clarisse *et al.,* 2002]), mostly in the speech processing context, no universal monophonic pitch detector has been developed. In fact, each of them has its own advantages and limitations: some focus on the problem of pitch detection in the singing voice (e.g., [Ryynänen, 2004; Viitaniemi *et al.,* 2003; Clarisse *et al.,* 2002; Haus and Pollastri, 2001; McNab *et al.,* 1996a]), others propose more robust solutions to the common octave-error problem (e.g., [de Cheveigné and Kawahara, 2002]), still others are designed for efficiency in order to cope with real-time needs (e.g., [de la Cuadra *et al.,* 2001]), and so forth.

Among these, pitch detection of the singing voice has proved to be a difficult problem, even in a monophonic context, mainly as a consequence of the acoustic properties of the human voice, which involves both voiced and unvoiced sounds. Vocal sounds can be classified as voiced or unvoiced (which can be further divided into fricative and plosive), based on their mode of excitation. Basically, voiced sounds correspond to vowels and give rise to periodic waveforms. Therefore, they enclose the performed musical pitches and are easier to analyze. On the other hand, unvoiced sounds relate to consonants (except for [m], [n] and [l], which are voiced, [Haus and Pollastri, 2001]). These are difficult to analyze because of their noise-like properties. Voiced sounds dominate during singing but, even so, unvoiced elements are relevant, since they frequently convey rhythmic aspects of the performance.

The existing approaches can be categorized in different ways. Namely, it is practical to cluster them according to the processing domain, where time and frequency-domain methods can be defined.

Anssi Klapuri recommends, however, a different categorization, where algorithms are clustered based on the way they handle spectral information. In this organization, *spectral location*, *spectral interval* and *unitary* categories are defined [Klapuri, 2004, pp. 23]. Briefly, the first class of methods looks for frequency partials at harmonic spectral locations, the second group considers the spectral intervals between partials and unitary algorithms provide a trade-off between both mechanisms. Representative variants from each class are introduced in the following paragraphs, complying with Klapuri's organization. Extensive overviews on monophonic pitch detection can be found, e.g., in [Gómez *et al.,* 2003; Gerhard, 2003; de Cheveigné and Kawahara, 2002].

A. Spectral Location Algorithms

As mentioned, this category encompasses the class of algorithms that look for frequency partials at specific spectral locations, namely strategies based on harmonic pattern matching or wave periodicity analysis methods. In the latter, despite the fact that the analysis is conducted in the time-domain, a corresponding spectral representation is mathematically implicit, where the locations of frequency partials are used.

Autocorrelation Function (ACF)

One of the most commonly used classes of time-domain algorithms is the one founded on the autocorrelation function. It consists on checking how similar a signal is to itself at each point, i.e., how well it superimposes with itself at different time lags. The ACF of a discrete signal $x[n]$ is usually defined as in (3.1):

$$r[\tau] = \frac{1}{N} \sum_{k=0}^{N-\tau-1} x[k] \cdot x[k+\tau] \tag{3.1}$$

where N denotes the duration of the signal in number of samples and $r(\tau)$ is the value of the autocorrelation function for a time lag τ. The fundamental frequency is then habitually determined as the maximum of $r(\tau)$ after zero lag (since the ACF has a maximum when the signal is compared to itself, i.e., at zero time lag).

The ACF can be more efficiently computed in the frequency domain via the Fast Fourier Transform (FFT) algorithm [Smith, 1997, pp. 225-242], according to (3.2). First, the signal is transformed into the frequency domain by the FFT and then the square of the magnitude spectrum is obtained and transformed back to the temporal domain. In (3.2) $|X[k]|$ denotes the magnitude of the spectrum at the k^{th} frequency bin.

$$r[\tau] = FFT^{-1} \left| FFT\left(x[n]\right) \right|^2 = FFT^{-1} \left| X[k] \right|^2 \tag{3.2}$$

In conceptual terms, the ACF as calculated by (3.2) differs from (3.1) in that the FFT-based ACF is equivalent to circular autocorrelation [Smith, 1997, pp. 184]. This corresponds to always using N values in the summation in (3.1), as a result of feeding in the initial values of x after the $N\text{-}1^{th}$ sample, i.e., in a circular way.

The ACF can be looked upon as a mechanism that accentuates frequency partials at harmonic locations of the magnitude spectrum. This becomes clearer if we rewrite (3.2) as follows, (3.3):

$$r[\tau] = \frac{1}{N} \sum_{k=0}^{N-1} \cos\left(\frac{2\pi\tau k}{N}\right) \left| X[k] \right|^2 \tag{3.3}$$

Basically, the previous equation says that when τ matches the true sound period the

square magnitude spectrum is maximally weighted at the harmonic locations.

ACF-based pitch detectors are relatively immune to noise, but are sensitive to formants and spectral peculiarities in the sound [Gómez *et al.,* 2003]. In effect, raising the magnitude spectrum to the second power emphasizes spectral peaks in relation to noise but aggravates the spectral peculiarities of the sound under analysis (e.g., strong formant structure, which amplifies the harmonics in some frequency regions).

Moreover, this class of methods has the drawback of being prone to "twice too low" octave errors. In reality, in temporal analysis, the signal is also periodic at multiples of the fundamental period. Such integer multiples (and doublings in particular, i.e., half the F0) have sometimes higher weights in the ACF producing the referred class of errors.

Examples of pitch detectors that resort to the ACF include the following: [Medan *et al.,* 1991], where the cross-correlation function over the range of feasible pitch values of synthetic and real speech data is maximized; [Talkin, 1995], which is based on a two-step calculation of the normalized cross-correlation function between successive segments of the input signal; or the YIN algorithm [de Cheveigné and Kawahara, 2002], where a number of modifications are introduced in order to decrease estimation errors [38] (e.g., deal with octave errors, increase pitch accuracy by interpolation) and to reduce the number of free parameters (e.g., upper frequency limits due to peaks near zero lag).

Cepstral Analysis

The cepstral[39] pitch detector, proposed by Michael Noll in 1967 [Noll, 1967] (cited in [Gómez *et al.,* 2003]) for pitch detection in speech signals has close model level similarities with the ACF. In fact, the cepstrum is computed as the inverse Fourier transform of the logarithm of the power spectrum of the signal, as in (3.4):

$$c[n] = FFT^{-1} \log\left(\left|FFT\left(x[n]\right)\right|\right) \tag{3.4}$$

The idea of taking the logarithm is to separate the source and transfer functions. Hence, the pulse sequence originating from the periodicity source reappears in the cepstrum as a strong peak at the fundamental quefrency.

By using the logarithm, cepstral analysis has the opposite benefits and limitations of ACF: it is reasonably robust for signals with strong formants and spectral peculiarities but is inaccurate in the presence of noise. As for "twice too low" octave errors, this obstacle is kept.

---

[38] Actually, one of the modifications was to replace the ACF by the related average magnitude-difference function.

[39] Most of the terms related to cepstral analysis are anagrams of frequency-domain terms, for example, cepstral instead of spectral, cepstrum instead of spectrum, quefrency instead of frequency, etc.

Harmonic Matching Methods

Harmonic matching algorithms rely on the peaks of the magnitude spectrum for F0 determination. In a musical context, the identified spectral peaks are compared to the predicted harmonics for each F0 candidate, from which a fitting measure is computed.

In [Maher and Beauchamp, 1993] a fitness measure designated as "Two-Way Mismatch" is described, where, for each F0 candidate, mismatches between the theoretical and the obtained harmonic frequencies are averaged over a fixed subset of the available partials. With the purpose of increasing the robustness of the method to noise or to the absence of certain partials, a weighting scheme is employed.

In the same category, Boris Doval and Xavier Rodet [Doval and Rodet, 1991] follow a probabilistic approach using a maximum likelihood spectral pattern matching pitch detector. The general idea is to look for the F0 that best explains the partials perceived in the magnitude spectrum. In this way, Gaussian functions centered on each multiple of a hypothesized F0 are used to represent the likelihood of observing the partials given the F0 candidate, in a Bayesian manner. This approach recurs to a number of random variables such as the fundamental frequency, the amplitude envelope, the presence or absence of specific harmonics, the probability density of specific partials and the number and probability of other partials and noise partials.

B. Spectral Interval Algorithms

The main shortcoming of spectral location algorithms is their inability to appropriately cope with inharmonic sounds. Indeed, by weighting spectral components according to their spectral locations, the sounds produced by real musical instruments are not properly dealt with, since the harmonics are not usually found at their exact theoretical places. Spectral interval methods are then proposed to overcome this difficulty.

As the name suggests, spectral interval algorithms are based on measuring the spectral intervals between frequency partials. These algorithms work relatively well for non-ideal harmonic sounds, since the intervals between harmonics remain more stable than their exact theoretical locations, for example in the spectra of piano sounds.

Such magnitude spectra can be regarded as being "periodic", in the sense that harmonic peaks appear at regular intervals. An obvious and common way of determining that period is to compute the ACF of the magnitude spectrum. Examples of algorithms derived from spectrum autocorrelation are [Lahat *et al.,* 1987; Kunieda *et al.,* 1996].

In terms of octave errors, unlike temporal autocorrelation, which is prone to twice too low F0 errors, spectrum autocorrelation frequently leads to F0 doubling errors. In fact, the magnitude spectrum is quasi-periodic at multiples of the F0. Thus, such integer multiples, and doublings in particular, may receive increased weight in the calculation of the ACF, causing the mentioned "twice too high" errors.

C. Unitary Model

A third category of algorithms is based on a trade-off between spectral location and spectral interval methods. Here, the underlying principle is to evaluate, not the periodicity of the time-domain waveform or of the magnitude spectrum, but the periodicity of the time-domain amplitude envelope.

Envelope Periodicity

In effect, signals with more than one frequency component reveal periodic fluctuations in the time-domain amplitude envelope, i.e., show beats. The rate of the observed beats is a function of the frequency difference of each pair of frequency components. In the case of harmonic signals, such as the ones from musical sounds, an interval corresponding to the fundamental period will dominate and so the F0 will be noticeable in the amplitude envelope of the signal.

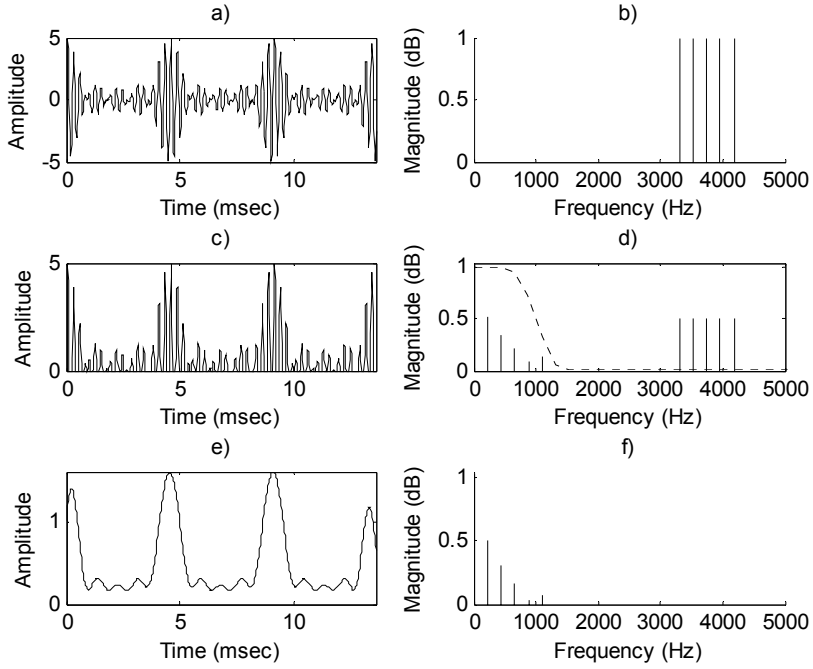


Figure 3.3. Envelope periodicity: a) and b) original time-domain signal and respective magnitude spectrum; c) and d) half-wave rectified signal and spectrum; e) and f) amplitude envelope and spectrum.

This is illustrated in Figure 3.3, for a signal composed of the 15th to 19th harmonics of a 220 Hz fundamental frequency tone (based in [Klapuri, 2004, pp. 27]). There, the

magnitude spectrum (panel b) of the original signal (panel a) contains information on the spectral locations of the harmonics of the initial signal. After half-wave rectification (panel c), the resulting spectrum encloses also information on the periodicity of the amplitude envelope (panel d), which is determined by the intervals in frequency between the original partials. Thus, spectral interval information is added to the former spectrum. Finally, the amplitude envelope (panel e) is obtained by low-pass filtering the rectified signal. The respective spectrum is depicted in panel f.

Envelope periodicity methods offer an elegant trade-off between spectral location and spectral interval information. The trade-off between the two types of information is determined by the characteristics of the low-pass filter (dashed line in Figure 3.3d): if the cutoff frequency is tuned in order that the initial frequency components are kept, a subsequent periodicity analysis utilizes both spectral location (from the first spectrum) and spectrum interval (from the amplitude envelope) information. Typically, the cutoff frequency is set to 1 kHz and a smooth transition band is usually defined so that signal components above 1 kHz are increasingly attenuated.

Auditory Models: Unitary Models

The spectral location/interval dialectic is present as well in several theories of pitch perception. Indeed, some philosophies base the calculation of the fundamental frequency on the locations of partials (attempting to find the fundamental of which they are harmonic), whereas others use the beats between them (the fundamental corresponding to the frequency of beating) [Bregman, 1990, pp. 236].

It is likely that human pitch perception resorts to both spectral locations and spectral intervals. In reality, whereas lower harmonics generally fall into separate critical frequency bands, several higher-frequency harmonics may belong to the same critical band[40]. The latter, designated as *unresolved harmonics*, interact with each other, producing beats in the previously described way. In this way, it is plausible that spectral locations are used in the lower parts of the spectrum, whereas, in the higher frequency channels, beat registration dominates [Bregman, 1990, pp. 237].

Moreover, some controversy also exists as to the procedures behind the detection of the frequency components in the input signal. Here, *place* (or frequency) theories use the evidence that different places in the basilar membrane of the inner ear respond maximally to different frequencies. On the other hand, *timing* (or periodicity) theories are based on the fact that the part of the basilar membrane that responds best to a given frequency component tends to vibrate at the frequency of that component as well [Bregman, 1990, pp. 235]. The auditory system could then use this information to infer

[40] Basically, for now it suffices to say that critical bands are channels for processing. Spectral components in different frequency ranges fall into different frequency channels and, thus, are separately processed [Hartmann, 1997, pp. 256]. This will further discussed in Section 3.3.1.

the spectrum of the input sound.

Implementations of each of the abovementioned theories yield different kinds of results and, hence, only explain parts of the problem. However, recent models of human pitch perception attempt to unify these competing psychoacoustic theories into one single model, able to reproduce a wide range of phenomena in human pitch perception (therefore the term "unitary model"). Researchers such as Raymond Meddis and colleagues [Meddis and O'Mard, 1997; Meddis and Hewitt, 1991] or Malcolm Slaney and Richard Lyon [Slaney and Lyon, 1990; Slaney and Lyon, 1993] have conducted considerable of work towards this goal, based on an early work by Joseph Licklider ([Licklider, 1951] (cited in [Slaney and Lyon, 1990]). This author was the first to propose correlograms as a framework for pitch perception.

In the unitary approach, both timing and place information are taken into consideration, through band-wise signal analysis followed by periodicity evaluation in each channel. The detected periodicities are then integrated across channels. Both spectral locations and intervals are used, since half-wave rectification is performed in each band, from which the spectral components of the amplitude envelope are added to the spectrum. The rectified signal in each band may (or not) be filtered, according to the desired trade-off between spectral location and spectral interval information.

Another advantage of unitary models is that, by separating the analysis into different frequency bands, increased robustness to corrupted signals can be achieved. For example, frequency bands with more favorable SNR might provide crucial information for pitch detection.

A focus of some criticism in these approaches concerns the fact that no different mechanisms are adopted for resolved and unresolved harmonics, as referred to in [Tolonen and Karjalainen, 2000]. Another drawback is that the execution of these models is usually computationally expensive, even though some of the processes carried out in the ear are frequently simplified. Moreover, controversial methods for periodicity calculation are employed, as the exact operations executed in the brain are still the subject of some controversy. Nonetheless, it is argued that the outcome reproduces quite faithfully the work performed by the human auditory system.

Further details on the general processing steps of auditory-model-based pitch detectors, and particularly of Slaney and Lyon's model, will be given in Section 3.3.

3.1.4. Polyphonic Pitch Detection

As for polyphonic pitch detection, the goal is to find the F0s of all sounds present in a mixture. In this case, many more difficulties are expected. Particularly, as a consequence of the presence of several instruments of both pitched and unpitched nature, problems

such as peak masking and spectral collisions increase its complexity level, in terms of the actual detection of pitches (namely regarding their precise frequencies and intensities). Moreover, concurrent sounds whose F0s are related by small integer ratios may cause the erroneous detection of nonexistent tones, e.g., the root tone of a chord. Spectral matching of tone models is also more complex, since the different instruments in the mixture have distinct and varying spectral properties.

The described difficulties are illustrated in Figure 3.4, where the magnitude spectrum of the saxophone sound depicted in Figure 3.1 is mixed with three other MIDI synthesized sounds (namely, piano, flute and violin). The obtained spectrum, depicted in panel b), is significantly more complex. The addition of percussion would increase the complexity level even more.

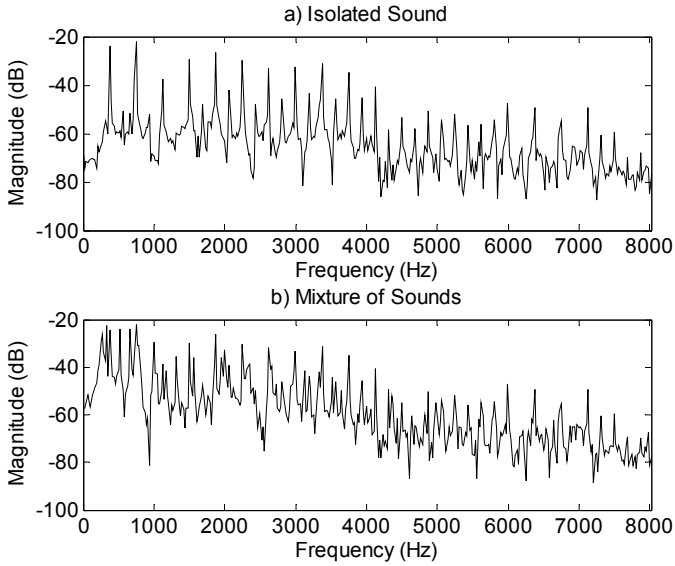


Figure 3.4. Spectra of an isolated harmonic sound and of a mixture of sounds.

In general, monophonic pitch detection algorithms are not adequate to polyphonic pitch detection tasks. Nevertheless, they can be adapted to simple polyphonic pitch detection problems [Gómez *et al.,* 2003], as well as melody detection in polyphonic contexts. A straightforward extension of standard monophonic pitch detectors, such as autocorrelation-based ones, consists of selecting more than one peak in each frame, along with post-processing for pruning. This is the approach we follow for melody detection, as will be seen. However, these extensions do not suit well the requirements of full music transcription, since they are a source of both ghost and missing notes, making it difficult to attain a good balance between over and under-detection. Also, such method-

ologies are not designed to cope with situations such as spectral overlapping and their effects, i.e., spectral collisions and peak masking.

Thus, dedicated polyphonic-oriented algorithms have been developed, mostly in the context of automatic music transcription systems. The devised mechanisms are even more diverse than the ones for single-pitch detection, approaching the problem from different perspectives, e.g., physiology of hearing, perceptual aspects involved in the listening process, instrument tone models, musicological principles, etc. A few representative polyphonic pitch detectors are described in the next paragraphs. More comprehensive overviews can be found in [Klapuri, 2004; Bello, 2003; Hainsworth, 2001].

The first attempts towards this goal date back to the 1970s, with the work by James Moorer [Moorer, 1977]. His system allowed a maximum of two simultaneous instruments, under heavy constraints: both instruments should be pitched, the dynamics were controlled (no glissando neither vibrato permitted), the two parts should not cross and simultaneous notes were only admitted as long as their fundamental frequencies and harmonics did not overlap. This last restriction is particularly difficult to deal with, since most common music intervals are small whole number ratios of each other, e.g., major thirds, fifths, octaves, etc. Such a limitation makes the addition of further instruments prohibitive. Hence, this initial effort was too restricting and, therefore, unpractical in a general framework.

Moorer's work was continued during the early and middle eighties by a group of researchers from the Center for Computer Research in Music and Acoustics (CCRMA), from Stanford University, e.g., [Chafe *et al.,* 1982; Chafe *et al.,* 1985; Chafe and Jaffe, 1986], for the transcription of acoustic piano signals. They proposed a set of heuristic rules to group peaks at the output of a filterbank (based on the bounded-Q transform).

Later on, Robert Maher also implemented an algorithm for the transcription of duets [Maher, 1989; Maher, 1990] (cited in [Klapuri, 2004, pp. 69]). There, frame-based spectral analysis was conducted, where the two F0s were selected as the pair that minimized the difference between the predicted and observed harmonics. However, in both Stanford and Maher's strategies, the polyphony was confined to two voices with no crossing of the F0s in each voice.

As part of the "Kansei" music system (which attempted to mimic the human response to music), Haruhiro Katayose and Seiji Inokuchi [Katayose and Inokuchi, 1989] built up a transcriber for guitar, piano and shamisen (a traditional Japanese instrument). There, peak extraction was carried out in the frequency domain and a number of heuristic rules were suggested to group the obtained peaks into notes. The transcriber was tested on polyphonies with more than two notes but the performance was poor in this case.

In 1993, Michael Hawley conformed to a different methodology to the transcription of polyphonic piano performances [Hawley, 1993]. His system was more flexible than

Christofer Chafe *et al.*'s, as more than two notes could be present simultaneously. It was based on a differential spectrum analysis, similar to taking the difference of two adjacent frames in an STFT. Note onsets were also looked for in the high-frequency region. The method was reported to being reasonably successful (although this is not clear, since no detailed tests are provided), demonstrating the viability of polyphonic transcription for a specific instrument such as piano. In reality, narrowing the focus of analysis to one particular instrument with well-known characteristics permits the relaxation of other constraints, leading to better accuracy. Nevertheless, good results are only attained with the specific instrument used, and so this solution still lacks generality.

Earlier automatic music transcription approaches were very limited in regard to the number of simultaneous allowed sounds, the defined pitch range and the relationships between concurrent sounds. Only in recent times, systems became reasonably accurate in higher polyphonies, without restricting the analysis to one single well-modeled instrument. Even so, a general-purpose, robust and reliable method of automatic transcription of polyphonic music is yet to be devised. Indeed, the performance of "modern" polyphonic pitch detection algorithms decreases progressively as the number of voices increases. In addition, the performance decreases substantially in the presence of noise.

Kunio Kashino and colleagues [Kashino *et al.,* 1995] created a system where sinusoidal tracks[41] were extracted from the input signal and clustered into note hypotheses, resorting to the implementation of some of the perceptual cues of sound organization described in Section 2.2, namely harmonicity and onset timing. Moreover, attempts towards the identification of the source of each note were conducted recurring to timbre models. Here, coinciding frequency components were resolved via pre-stored tone memories. In terms of musical knowledge, chord note relations and statistics of chord transitions were employed. The integration of top-down and bottom-up information was accomplished with recourse to a Bayesian probability network, conceptually based on a blackboard model (see next paragraph). The evaluation set consisted of random mixtures of samples from five different instruments, forming polyphonies with up to three simultaneous voices.

Keith Martin [Martin, 1996] also suggested a strategy founded on blackboard frameworks. The blackboard system is composed of a global database (where hypotheses are proposed and developed), a scheduler (that monitors and controls the interactivity within the system), and knowledge sources (corresponding to experts). The blackboard made use of knowledge about principles from physical sound production, rules governing tonal music and "garbage collection" heuristics. Namely, the probabilities of occurrence of different notes (either concurrent or sequentially) can be estimated based on

[41] Basically, sinusoidal tracks are formed by peak detection in each frame, followed by peak continuation, which connects peak candidates with similar frequency values in consecutive frames. This is further developed in Chapter 4.

databases of written music. Additionally, F0s in particular intervallic relations were favored by the used musical rules. This system employs an auditory model, where a log-lag correlogram was used as a mid-level representation. One drawback of Martin's approach is that it was simulated using only a short musical excerpt, specifically, a piano performance of a four-voice Bach chorale. Similarly, Juan Bello also recurred to the blackboard framework in the analysis of simple polyphonic music [Bello, 2003]. There, piano pieces by well-known composers were used for evaluation.

Another work that makes extensive use of perceptual grouping principles is the one developed by Andrew Sterian [Sterian, 1999]. There, sinusoidal tracks are utilized as mid-level representation. Perceptual grouping rules are then represented as a set of likelihood functions that evaluate the probability of the observed partials given a hypothesized grouping. The defined likelihood functions take into account cues such as harmonicity, onset and ending timings, partial density, among others. A multiple-hypothesis tracking method is then used to find a solution (sub-optimal in this case) for the best grouping. The system was evaluated in a small test set with up to four simultaneous sounds and the authors report a note recognition index for the recorded song excerpts of around 0.5 (computed as a function of the percentage of true notes captured and false positives).

Luís Martins used as well sinusoidal tracks as mid-level representation, with a slight yet important variation [Martins, 2001]: instead of performing peak continuation on the total set of peaks in each frame, harmonic structures are first identified according to the harmonic relations between peaks; then, peak continuation is conducted on the detected harmonic structures. The constructed trajectories are accepted or rejected based on a clustering and pruning algorithm, which, among other tasks, attempts to eliminate ghost trajectories that result from harmonic relationships. The system was tested on MIDI synthesized sounds with polyphonies of up to three simultaneous notes with reasonable success. More difficulties were encountered while transcribing a recording of a piano piece.

Alain de Cheveigné and Hideki Kawahara [de Cheveigné and Kawahara, 1999] extended the auditory-model-based pitch detector proposed by Raymond Meddis and Michael Hewitt [Meddis and Hewitt, 1991] to the multi-pitch case, where an iterative cancellation-detection scheme was suggested. First, a pitch is detected and the corresponding sound is cancelled. Pitch detection is then repeated for the residual sound until all pitches are determined. Sound cancellation is carried out either by channel selection or by means of within channel cancellation filtering. Despite the fact that the dataset adopted for evaluation was quite artificial (e.g., a maximum of three concurrent perfectly periodic signals, with a pitch range less than an octave), the iterative methodology appears to work successfully.

Meddis and Hewitt's model also inspired the multiple-pitch detector designed by Tero Tolonen and Matti Karjalainen [Tolonen and Karjalainen, 2000]. The central idea of the authors was to devise a computationally efficient version of the initial algorithm,

adapted to polyphonic pitch detection. To this end, only two frequency bands are used instead of the original 40 to 120 channels. Spectral flattening is performed by inverse warped-linear-prediction filtering and the trade-off between robustness to noise and to spectral peculiarities is addressed by applying of a generalized ACF function. The summary autocorrelation function is then enhanced so that spurious peaks are discarded. However, although several of such peaks are disposed of, both false positives and false negatives occur. The algorithm was tested with noisy and clean musical chords, as well as mixed speech signals, showing reasonable accuracy.

Band-wise processing was also conducted by Anssi Klapuri [Klapuri, 2003]. Nonetheless, this method is less computer-intensive, given that only a single STFT per frame is needed, after which local regions of the spectrum are separately processed. Namely, 18 logarithmically distributed bands from 50 Hz to 6 kHz are used, each spanning a 2/3-octave wide region of the spectrum that is weighted with a triangular frequency response. Then, a fundamental frequency likelihood vector is calculated at each band. These likelihoods are combined to yield a global pitch likelihood, in a manner capable of handling inharmonicities. An iterative estimation and cancellation procedure is then implemented. First, the most likely pitch is selected as the predominant pitch and its partials are subtracted from the mixture (based on the spectral smoothness principle). Then, pitch detection and sound cancellation are applied iteratively to the residual. Furthermore, the method encloses mechanisms for noise suppression inspired on RASTA processing and for the estimation of the number of concurrent sounds in the signal. The system was tested on mixtures of notes (one up to six), and proved, for this particular set, to be able to resolve at least the most prominent pitches in the mixture. Moreover, the author reports results that surpass the average of ten trained musicians in musical chord identification tasks.

Klapuri also proposed an auditory-model-based system for multiple-pitch detection [Klapuri, 2005], which was then employed by himself and Ryynänen in a polyphonic pitch detector [Ryynänen and Klapuri, 2005a]. This system is a super-set of the one created by the same authors for melody detection [Ryynänen and Klapuri, 2005b], and described in Section 2.4.2. As referred to before, a note event HMM, a silence model and a musicological model are utilized for polyphonic transcription. However, instead of selecting only the optimal path through the network formed by note event and silence models, several paths are looked for. Namely, after finding the optimal path via the Token-passing algorithm, the used note models are removed, the observation likelihoods for the silence model are recalculated and the next best path is looked for. This is repeated for a pre-defined number of paths until the found paths contain no more notes.

Although the recent approaches are more general and flexible than Moorer's early system, a robust and reliable method of automatic transcription of polyphonic music is yet to be conceived. In reality, for practical purposes, automatic transcription in an arbitrary context is still far from being solved. This is demonstrated by the attempted efforts

towards the development of commercial products. Such systems aim to convert polyphonic music recordings to the MIDI format but, despite their usefulness for simple polyphonic music, results are still insufficient for "real-world" music.

3.2. Pre-Processing: RASTA Processing

As discussed in Section 3.1.2, the objective of the pre-processor is to perform data reduction so as to facilitate F0 extraction. In this way, noise suppression and enhancement of features useful for F0 determination are important tasks in this respect. The problem of acoustic noise suppression was studied mainly in the speech-processing domain (see [Hess, 1983; Klapuri, 2004, pp. 80] for a review of such techniques).

Basically, the sound generated by physical vibrators is first filtered by the resonance structures (e.g., body of a guitar, characteristics of the human vocal tract) and by the environment (reflections in walls, etc.) and then linearly superimposed with other simultaneous sounds and noise. The first type is named *convolutive noise* and the second one is *additive noise*. The underlying F0 is best revealed if both are removed.

The suppression of convolutive noise is usually denominated *spectral whitening* (or flattening), since it aims to normalize the spectrum peculiarities of the sound source and the environment, i.e., removing "color" from the spectrum, but leaving the spectral fine structure intact. A common way to accomplish it is by inverse linear predictive filtering. Namely, in the music context, Tolonen and Ellis carry out spectral whitening, respectively by inverse filtering with warped linear-prediction [Tolonen and Karjalainen, 2000] and by normalizing the powers of the outputs of a band-pass filterbank [Ellis, 1996, pp. 77]. In our work, we employ an auditory model, whose signal compression is often regarded as similar to spectral whitening, as will be seen in Section 3.3.1.

As for additive noise, when analyzing speech sounds it is normally assumed that background noise characteristics are slowly-varying in comparison to the target speech signal. Hence, a typical way of removing it consists of first estimating the noise spectrum over a longer period of time and then subtracting the noise component from the mixture.

However, the concept of additive noise is different in music. Here, such "noise" comes from the presence of percussive instruments (particularly drums), which are short in duration and transient in nature, unlike the slowly-varying continuous noise usually assumed in speech. In fact, continuous noise is not typical of musical signals and, thus, its estimation over a long window is not adequate. In our work, we consider everything that is not part of the melody, i.e., all sorts of accompaniments, either pitched or unpitched, as additive noise. Particularly, percussive components should be suppressed since they lead to low SNR and are consequently a major source of peak masking (al-

though it is very difficult to achieve it in practice). Therefore, the notion of additive noise is entirely different in speech and music, and so its suppression should obey to different criteria.

In this way, specific pre-processing mechanisms for music are necessary. Nevertheless, we are not aware of many relevant studies under this topic. One notable exception is the strategy pursued by Klapuri [Klapuri, 2003], based on the principles of RASTA (i.e., RelAtive SpecTrAl) processing [Hermansky *et al.*, 1993].

The idea of the algorithm is to remove both convolutive and additive noise simultaneously and this is conducted in each analysis frame due to the transient nature of percussive sounds. Hence, the method relies on a signal model for harmonic sounds containing both kinds of noise, according to (3.5):

$$X[k] = S[k] \cdot H[k] + N[k] = X_H[k] + N[k] \tag{3.5}$$

In the previous expression, $X[k]$ is the observed power spectrum of a discrete input signal and $N[k]$ is the power spectrum of unknown additive noise, here represented by all non-harmonic components. This model assumes that the additive noise and the signal are uncorrelated. Furthermore, $N[k]$ cannot be assumed stationary.

Still in (3.5), $S[k]$ denotes the power spectrum of the vibrating system whose fundamental frequency should be measured (for example, a guitar string). This spectrum is filtered by $H[k]$, which represents the frequency response of the body of the musical instrument, the operating environment and other convolutive noise, as denoted by $X_H[k]$.

Convolutive noise is eliminated by magnitude warping the power spectrum of the signal, which equalizes $X_H[k]$ and allows the linear subtraction of the additive noise from the result. This is accomplished following the lines of RASTA processing, as in (3.6):

$$Y[k] = \ln\left(1 + \frac{1}{g} X[k]\right) \tag{3.6}$$

There, g is a scaling factor that is adaptively calculated in each analysis frame, acting to scale the level of the additive noise floor to a value close to unity. Moreover, it is assumed that the amplitudes of the important frequency partials are well above the additive noise floor and that the majority of frequency components come from noise, rather than from the spectral peaks.

The performed logarithmic operation has some interesting effects. Indeed, given that the additive noise is low after scaling, it goes through a linear-like transform and remains additive. On the other hand, in conformity with the assumption that important frequency partials are clearly above the additive noise floor, the spectral peaks go through a logarithmic-like transform. Thus, the spectrum is flattened and so spectral peculiarities are attenuated much in the same way as in cepstral processing (as a consequence of the

logarithmic operation).

Developing (3.6) a bit more, it turns out (3.7), where, M[*k*] denotes the magnitude-warped additive noise.

$$Y[k] \approx M[k] + \ln\left(1 + \frac{1}{g} X_H[k]\right) \tag{3.7}$$

The author has found that an optimal value of *g* depends on the level of both the additive noise and the spectral peaks. After experimenting with different models, Klapuri concluded that the best performance was achieved by averaging the power spectrum in the frequency range of interest via the cubic root, as in (3.8). There, indices k_0 and k_1 are determined based on the employed frequency range, corresponding, respectively to frequencies of 50 Hz and 6.0 kHz.

$$g = \left(\frac{1}{k_1 - k_0 + 1} \cdot \sum_{k=k_0}^{k_1} \left(X[k]\right)^{1/3}\right)^3 \tag{3.8}$$

The additive noise component, M[*k*] in (3.7), is then suppressed by applying a specific spectral subtraction on Y[*k*]. The noise is first estimated by computing a moving average $\hat{M}[k]$ over Y[*k*] on a logarithmic frequency scale. The idea is that local average values of the spectrum represent the additive noise floor. Therefore, filtering the spectrum is this fashion should result in an estimate of the noise floor. More specifically, the magnitude of M[*k*] for $k = k_i$ is obtained by calculating a Hamming window weighted average over the values of Y[*k*] around k_i. The width, *W*, of the Hamming window depends on the center frequency *f* corresponding to k_i, according to (3.9):

$$W(f) = \beta \cdot 24.7 \cdot \left(4.37\frac{f}{1000} + 1\right) \tag{3.9}$$

In the previous expression, the width of the window is $\beta = 4.8$ times the width of an Equivalent-Rectangular Bandwidth critical-band (see [Hartmann, 1997, pp. 245-246]).

The estimated noise spectrum, $\hat{M}[k]$, is then linearly subtracted from the magnitude-warped power spectrum, Y[*k*], where negative values are set to zero, as in (3.10):

$$Z[k] = \max\left(0, Y[k] - \hat{M}[k]\right) \approx \max\left(0, \ln\left(1 + \frac{1}{g} X_H[k]\right)\right) \tag{3.10}$$

The executed procedures are illustrated in Figure 3.5, for an excerpt with two harmonic sounds and a snare drum. Panel a) shows the scaled power spectrum of the signal, panel b) depicts the warped-magnitude spectrum and the bottom panel represents the

spectrum after the subtraction of additive noise. As desired, after scaling, the noise floor is close to unity value and the spectral peaks are clearly above that value (top panel).

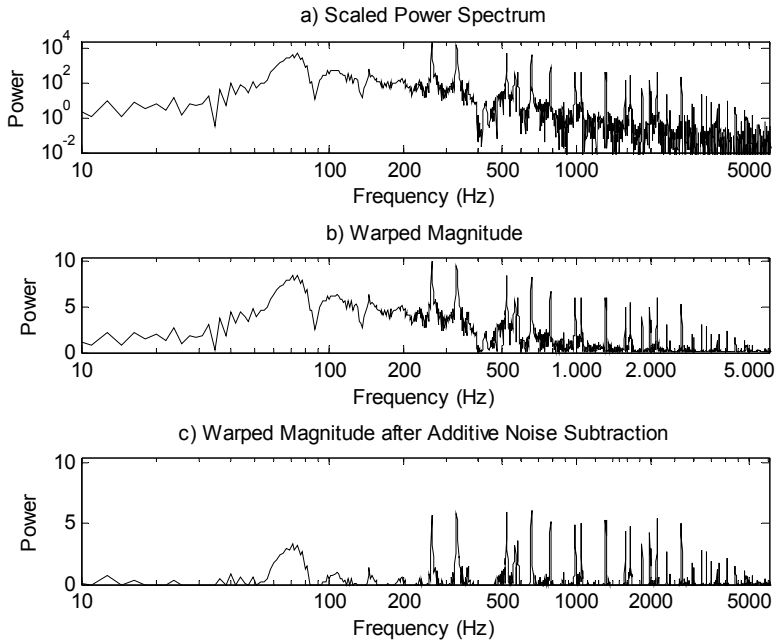


Figure 3.5. Additive noise suppression by spectral subtraction.

Finally, in case pitch detection is conducted on the spectral domain, the enhanced spectrum, $Z[k]$ in (3.10), is directly used. On the other hand, if pitch detection is performed in the temporal domain, the spectrum must be transformed back to the time domain. In this way, the obtained power spectrum is inverted to the time domain by the inverse Fourier transform. The phase of the original spectrum is employed in the inversion, after a clarification provided by the author in an e-mail. The resulting filtered frame data is then used for pitch detection as described in the next section. This is illustrated in Figure 3.6, for the same example.

There, we can see that the intensity level corresponding to the moment where the drum is hit (at around 200-300 msec) is considerably attenuated. We resynthesized the filtered signal, which confirmed the almost total removal of the drum. However, the output signal sounded somewhat "blurred". Resorting to visual image analysis, we could say that the sound lost contrast.

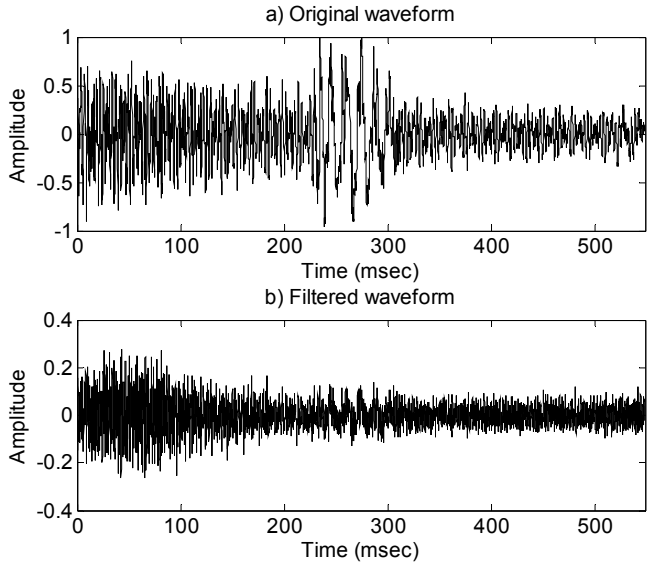


Figure 3.6. Filtered temporal signal.

3.3. Extraction: Auditory-Model-based Pitch Detector

Regarding pitch detection, the first impression we could possibly have is that, since we are only interested in the melodic line in an ensemble, and given our assumption that melody is usually dominant in the mixture, we could regard our problem as a single-pitch detection task in a "noisy" environment.

However, in practical situations melodic pitches are not always the most intense or the most likely ones. In reality, it is frequent that F0s corresponding to the periodicities of simultaneous notes may compete with each other and be alternately selected as the "best" pitch. Therefore, selecting several pitch candidates would make it possible to recover from situations where the F0 of the melody is not selected. In short, the idea of finding several peaks is motivated by the fact that missing notes cannot be recovered afterwards but, instead, false candidates can be eliminated in later stages.

Furthermore, we propose that in our work it is not essential to acquire the whole set of F0s within each frame, but rather the ones that are more likely to bear melodic information. In effect, detecting all the pitches present would give rise to irrelevant information in the context of melody detection. Also, the higher the number of notes we come up with, the more difficult it will be to select the ones that convey the main melodic line.

Putting the focus on the melody regardless of the other sources present, we follow a

melody-oriented multiple-pitch detection approach, whose basic idea is to capture only the most significant F0s in the context of melody detection, which we assume to be the most intense ones. Thus, despite the fact that monophonic pitch detectors are not adequate to polyphonic pitch detection tasks in general, we hypothesize that adapted single-pitch methods could suit well our needs in terms of melodic pitch detection accuracy, besides being conceptually simpler than polyphonic pitch detectors designed for full music transcription.

Namely, we decided to employ an Auditory-Model-based Pitch Detector (AMPD), given its improved performance compared to other evaluated pitch detectors[42] and the described benefits of the so-called unitary model. Additionally, despite the simple periodicity detection scheme carried out in our work, these models can be adapted to polyphonic pitch detection, e.g., by iterative estimation and cancellation schemes [Klapuri, 2005; Klapuri, 2004; de Cheveigné and Kawahara, 1999]. Moreover, we thought it advantageous to experiment with an auditory model, given the wide consensus about many of the processing mechanisms that occur in the physiological and more peripheral parts of the human auditory system. Hence, such a frequency analyzer would be expected to behave reasonably well. Indeed, this algorithm proved to work better than the other evaluated strategies, as will be seen in Section 3.6. However, one important drawback of this approach is that it is computationally expensive.

The AMPD, sketched in Figure 3.7, is based on Slaney and Lyon's auditory model [Slaney and Lyon, 1993]. It receives as input a raw musical signal (monaural, any sampling frequency, f_s, though only 22050 and 44100 Hz were used, and 16 bits quantization) and outputs a set of pitch candidates and their respective saliences.

Our goal is to collect pitch candidates at each time instant. Since we cannot define instantaneous time in a computational model, we have to use some sort of time granularity. Therefore, we select a small enough time window and conduct sound wave analysis in a frame-based manner. Here, we specify a 46.44 msec frame length and a hop size of 5.8 msec[43]. This window size constitutes a good trade-off between time and frequency resolution: it is small enough for the assumption of signal stationarity and large enough for accurate detection of pitches above 43.1 Hz (since each frame contains at least two periods of a sound wave with fundamental period equal to 23.22 msec). The defined hop size allows for a smooth transition between frames.

After dividing the musical signal into frames, we implement an auditory-model-based analysis of each frame, in order to detect the most salient pitches in each. This analysis

[42] The evaluated pitch detectors are described in Appendix A. Namely, we compared different kinds of approaches, based on spectral, autocorrelation, spectral autocorrelation and probabilistic analyses.

[43] These values were suggested during the ISMIR'2004 Melody Extraction Contest, leading to a window length of 2048 samples and a hop size of 256 samples (assuming f_s = 44.1 kHz). The window length was intentionally set to a power of 2 for FFT efficiency.

comprises four stages, diagrammed in Figure 3.7:

i) conversion of the sound waveform into auditory nerve responses for each frequency channel (defined in Section 3.3.1), using a model of the ear with particular emphasis on the cochlea, resulting a so-called cochleagram;

ii) analysis of the periodicities in each frequency channel using autocorrelation, from which a correlogram is obtained;

iii) determination of the global periodicities in the sound waveform by calculation of a summary correlogram, or summary ACF (SACF);

iv) detection of the pitch candidates in the frame by looking for the most salient peaks in the SACF.

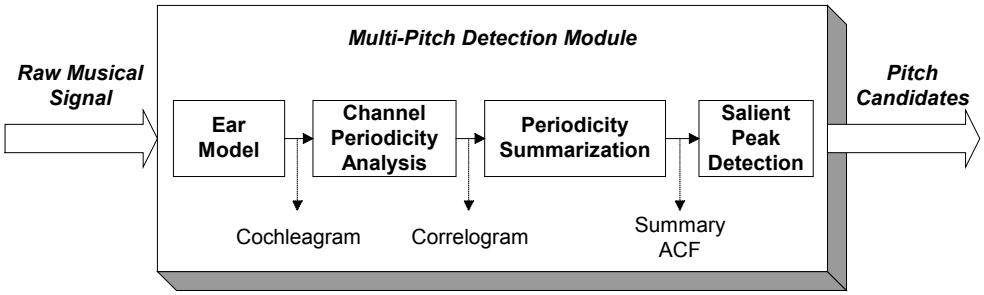


Figure 3.7. Auditory-model based pitch detector (AMPD).

The first two stages correspond to the original algorithm. As for periodicity summarization, unlike the original method, we do not normalize the accomplished SACF (discussed in the next subsection). Regarding salient peak detection, our procedure is also different, since several pitch candidates are identified instead of only the highest one.

3.3.1. Ear Model

In the first phase of the multi-pitch detection system, a model of the ear is implemented, which aims to mimic the tasks conducted by the outer, middle and, particularly, the inner ear in the first stages of auditory processing.

Before describing the adopted model, the general functioning of the cochlea and the competing theories of pitch perception and their consequences for the proposed representation are briefly presented.

A. The Cochlea

In the inner ear, the cochlea encodes information in the sound wave into a multi-channel representation of auditory nerve firing patterns. The output of the cochlear model is a two-dimensional representation of a sound waveform that allows its visualization as a time-frequency image. In this image, termed *cochleagram*, each line contains information about the auditory nerve responses for the corresponding cochlear (or frequency) channel (see Figure 3.13, on page 93). A cochleagram is then a measure of the way the frequency content of the signal changes over time, much in the same way as a conventional spectrogram is, despite several conceptual and practical differences. A good review of the tasks carried out in the cochlea and auditory nerve can be found in [Handel, 1989; Hartmann, 1997].

In short, the cochlea is a coiled, fluid-filled organ of the inner ear that is responsible for the transformation of the middle ear fluid vibration into neural firings [Handel, 1989, pp. 468]. Two elastic membranes divide it: the Reissner's membrane and the basilar membrane. It is the movement of the basilar membrane that is relevant for the generation of neural impulses. In effect, on top of the basilar membrane lies the organ of Corti, which contains the sensory cells, denominated hair cells. Due to the coiled-shape of the cochlea, some cells are located along the inside curve whereas others are located along the outside curve. The former are named inner hair cells and the latter, outer hair cells. As the air vibration reaches the inner ear (through the vibrations of the oval window), it creates a pressure wave in the fluid that fills the cochlea, which in turn distorts the basilar membrane. This distortion bends the hair cells, inducing their firing. Since these receptor cells converge to fibers in the auditory nerve, this causes the firing of neurons running towards the brain cortex. The are many more outer cells (about 12000) than inner cells (roughly 3500), but, curiously, 90 to 95% of the fibers of the auditory nerve are connected to the inner hair cells. Consequently, these are the most important ones to consider, as far as the modeling of nerve firing patterns is concerned.

One important property of the basilar membrane is that it is stiffer near the oval window (i.e., the *base*) and becomes more flexible towards the opposite end (i.e., the *apex*). Hence, it resonates close to the base for high frequencies, where it is stiff, and close to the apex for low frequencies, where it is more flexible, as illustrated in Figure 3.8. Thus, the basilar membrane acts as a frequency analyzer, since different points along it undergo maximum displacement as a function of frequency. Furthermore, a given frequency displaces obviously more than a single point along the basilar membrane. The displacement envelope is asymmetrical, being steepest towards the apex [Handel, 1989, pp. 476]. Therefore, for a given stimulation frequency, the amount of displacement leads to different excitation of the hair cells, according to their place, being maximum for the cells that match the stimulation frequency and spreading in the direction of the base.

This explanation for the behavior of the basilar membrane is called *place theory*, as mentioned in Section 3.1.3. It proposes a tonotopic organization of the auditory system,

i.e., an organization where different frequencies excite different hair cells [Hartmann, 1997, pp. 6].

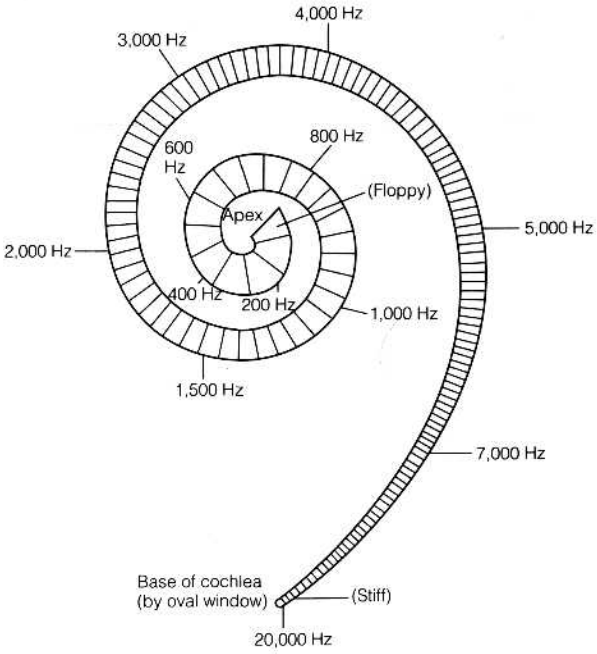


Figure 3.8. Frequency sensitiveness along the basilar membrane[44].

The place theory says basically that the excitation of each hair cell depends on its transfer function. Each cell has a best frequency for which a maximum firing rate occurs, responding to other frequencies in its vicinity with a lower rate of firing. In other words, each cell acts as a band-pass filter. Due to the asymmetry of the displacement envelope, the filters are asymmetrical too, with high-frequency slopes steeper than the lower ones [Hartmann, 1997, pp. 248]. Furthermore, the stiffness of the basilar membrane decreases nearly exponentially towards the apex. In this way, filter's center frequencies roughly follow a logarithmic scale. This description is illustrated in Figure 3.10 (page 91) for the cochlear (or auditory) filterbank described in [Slaney and Lyon, 1990].

Another common psychoacoustic explanation of the behavior of the basilar membrane is given by the timing theory (Section 3.1.3). This theory exploits the fact that the part of the basilar membrane that responds best to a given frequency component also tends to vibrate at the frequency of that component [Bregman, 1990, pp. 235]. Each movement originates a neural firing and, consequently, the frequency is coded directly

[44] Extracted from http://www.ai.rug.nl/~tjeerd/CPSP/docs/cochleaModel.html.

by the firing rate, e.g., a 400 Hz tone causes to hair cells firing at 400 times per second [Handel, 1989, pp. 472]. The auditory system could then use this information to infer the spectrum of the input sound. However, above 5 kHz neurons do not maintain their synchrony with the stimulus, which suggests the existence of other forms of representation for this situation.

Despite some competition between these two theoretic branches, it is more probable that a combination of the two is actually carried out. Indeed, the timing theory seems to dominate at frequencies up to 4 or 5 kHz, whereas higher frequencies are probably handled according to the place theory [Hartmann, 1997, pp. 294]. More information on these two hypotheses can be found in [Bregman, 1990; Handel, 1989; Hartmann, 1997].

B. Lyon's Ear Model

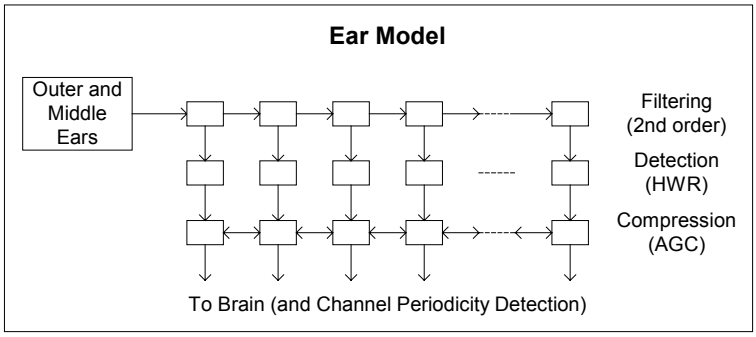


Figure 3.9. Lyon's ear model.

We use the ear model devised by Richard Lyon [Lyon, 1982] and programmed by Malcolm Slaney [Slaney, 1988; Slaney, 1998]. We give a short description of Lyon's model in the paragraphs below. For a more comprehensive analysis, we refer the reader to [Slaney, 1988; Slaney and Lyon, 1993]. Apart from this model, an extensive review of approaches for auditory modeling can be found in [Perdigão, 1997] (in Portuguese).

The model performs three main tasks: filtering, detection and compression, depicted in Figure 3.9 (adapted from [Slaney and Lyon, 1990]).

Filtering

First, the outer and middle ears add a slight high pass response to the system. These are modeled with a second order high-pass filter with a cutoff frequency of 300 Hz and unity gain at a quarter the Nyquist frequency, f_N. Further details can be obtained in [Slaney, 1988, pp. 22-25].

As for the cochlear filters, these model sound propagation down the basilar membrane, which behaves as a frequency analyzer. In this way, each filter corresponds to a cochlear channel that best responds to a particular frequency range.

The cochlear model described in [Lyon, 1982] combines a series of notch filters, which model the traveling pressure waves, with resonators, which model the conversion of pressure waves into basilar membrane motion or velocity [Slaney, 1988, pp. 8]. At each point in the cochlea, the acoustic wave is filtered by a notch filter. Each of these operates at successively lower frequencies in order that the net effect is to low pass the pressure wave. Basically, sound travels down the line of notch filters, being filtered at lower and lower frequencies. At the same time, resonators pick out a small range of the traveling energy and model the conversion into basilar membrane motion. It is this motion that is detected by the inner hair cells.

In the latest implementation of the model, the notch and resonator in each stage are combined. Hence, the poles in the resulting filter are set to the resonant frequency, so as to give a slight peak in the filter's response at that frequency, given the specified Q, whereas the zeros are placed slightly above to provide the band rejection.

The bandwidth of each filter is a function of its center frequency. At high frequencies the bandwidth, *bw*, is approximately equal to the center frequency, *cf*, divided by the filter's quality factor, i.e., filter Q, whereas at lower frequencies the bandwidth is nearly constant, i.e., a break frequency, *bf*, divided by Q [Slaney, 1988, pp. 11]. Formally, it comes (3.11):

$$bw[k] = \frac{\sqrt{cf[k]^2 + bf^2}}{Q}, \quad k = 1,2,\cdots,N \tag{3.11}$$

In the previous expression, *k* denotes the indices of the frequency channel, in a total of *N*, the ear break frequency, *bf*, is assigned a value of 1000 Hz and a Q factor of 8 is specified (suggested and kept default model parameters). The bandwidth calculated as described corresponds roughly to a critical band.

Each of these filters is overlapped by a fraction of the bandwidth, where an ear step factor, *sf*, equal to 0.25 is recommended by the authors. The top frequency, cf_{top}, slightly below the Nyquist frequency, f_N, is the reference from which the other center frequencies are defined. Formally, it turns out (3.12):

$$cf_{top} = cf[1] = f_N - \frac{\sqrt{f_N{}^2 + bf^2}}{Q} \cdot sf \cdot (zo - 1) \tag{3.12}$$

In the previous expression, the use of channel index 1 comes from the fact that lower channels relate to higher frequencies, as these resonate close to the base. As for the

ear zero offset (*zo*) parameter, this is described below (see Equation (3.15)).

In order to determine the center frequencies for all channels, the number of channels must be calculated. This is accomplished by finding the place, f_{low}, where the cascade pole Q is below 0.5. The number of channels, *N*, is then obtained as follows, (3.13):

$$f_{low} = \frac{bf}{\sqrt{4Q^2 - 1}}$$
$$N = \left\lfloor \frac{Q \cdot \left(-\log\left(f_{low} + \sqrt{{f_{low}}^2 + bf^2} \right) + \log\left(cf_{top} + \sqrt{{cf_{top}}^2 + bf^2} \right) \right)}{sf} \right\rfloor \tag{3.13}$$

Starting from the top center frequency, we step down in frequency by *sf* (step factor) times the bandwidth of the filter at the previous frequency, (3.14). Thus, center frequencies decrease exponentially from the base (index 1) towards the apex (index *N*), in accordance with the human ear physiology.

$$cf[k] = cf[k-1] - sf \cdot bw[k-1], \quad k = 2, 3, \cdots, N \tag{3.14}$$

As mentioned before, this model combines a series of notch and resonator filters. In the resulting second-order BPF, the poles are set to the center frequency and the zeros are positioned slightly above. This is summarized in (3.15):

$$\begin{aligned} zero_{\mathrm{CF}}[k] &= cf[k] + bw[k] \cdot sf \cdot zo \\ zero_{\mathrm{Q}}[k] &= sh \cdot \frac{zero_{\mathrm{CF}}[k]}{bw[k]} \\ pole_{\mathrm{CF}}[k] &= cf[k] \\ pole_{\mathrm{Q}}[k] &= sh \cdot \frac{cf[k]}{bw[k]} \end{aligned} \quad , \quad k = 1, 2, \cdots, N \tag{3.15}$$

There, the zero offset parameter, *zo* = 1.5, represents how far apart the zero is from the center frequency of the filter, i.e., the pole, and ear sharpness, *sh* = 5, denotes how much sharper the notch (zero) should be compared to the resonator (pole) [Slaney, 1988, pp. 12]. The defined zeros and poles are then used to design the desired second-order filters.

In our implementation, 118 and 96 cochlear filters result for sampling frequencies of 44100 and 22050 Hz, respectively. Figure 3.10 depicts the cascaded response for every 5th notch-resonator filter, using a logarithmic frequency axis (f_s = 22050 Hz)). This figure was created using Slaney's Auditory Toolbox, in whose source code we based our AMPD [Slaney, 1998].

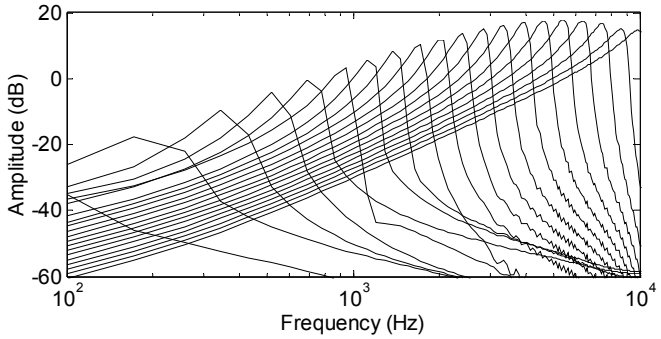


Figure 3.10. Frequency response of cochlear filters.

Detection

After filtering, the movements of the basilar membrane are converted into auditory nerve responses by the inner hair cells and the neurons of the auditory nerve. Since the inner hair cells only respond to movements in one direction, an array of half-wave rectifiers is employed to model the detection non-linearity of the hair cells, ensuring a non-negative output that can be used to represent neural response. This is a simple model of detection that, namely, does not account for saturation effects that occur when the motion is too large. Other more realistic models make use of soft-saturating half-wave rectification, local adaptation, refractory times and controlled firing rates, as referred to in [Slaney and Lyon, 1993].

Compression

Finally, four stages of multiplicative automatic gain control (AGC) compress the dynamic range of the input into a limited level that the auditory nerve can deal with. The automatic gain control is, in reality, a model of ear's adaptation: the response to a constant stimulus is first large and then, as the auditory system adapts to the stimulus, the response becomes smaller. Such adaptation is implemented as a variable gain, which attempts to keep the output of the AGC in each stage from exceeding a fixed level. To a certain extent, this normalization of hair cell activity is functionally similar to spectral flattening [Tolonen and Karjalainen, 2000].

To simulate adaptation, the AGC first operates at a point where it is sensitive to new sounds. After a loud sound is detected, the gain is turned down. The conducted procedures are sketched in Figure 3.11 (adapted from [Slaney and Lyon, 1993]).

There, z^{-1} denotes unit delay and the loop with feedback gain $(1 - \varepsilon) / 3$ represents a simple low-pass filter with a time constant defined in the ε parameter. There, the division by three takes the average output from the input, left and right channels. The target

parameter, *T*, is used to scale the input to the loop filter. Four AGC stages like the ones in Figure 3.11 are combined to model the range of adaptation rates present in the auditory system. In each stage, target values of 0.0032, 0.0016, 0.0008 and 0.0004 and time constants of 640, 160, 40 and 10 msec are respectively used. Figure 3.12 illustrates the output after four stages of AGC (solid line) for a constant input of 0.008 (dashed line).

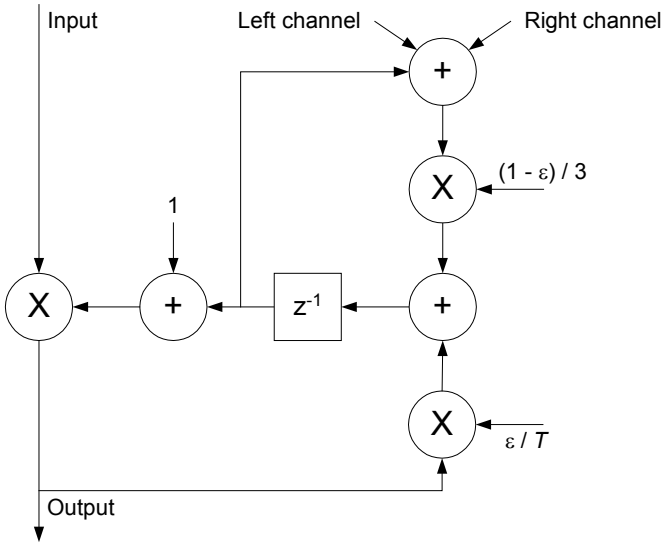


Figure 3.11. Automatic gain control.

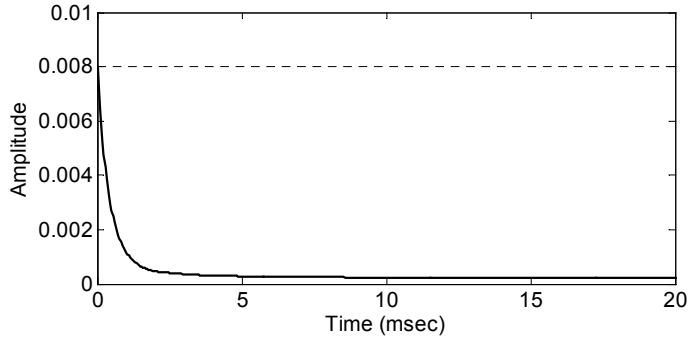


Figure 3.12. Output after four stages of AGC.

To sum up the ear model step, Figure 3.13 presents the resulting cochleagram for a monophonic saxophone riff (sampled at 22050 Hz) after filtering, detection and com-

pression. This simple example was chosen for ease of illustration.

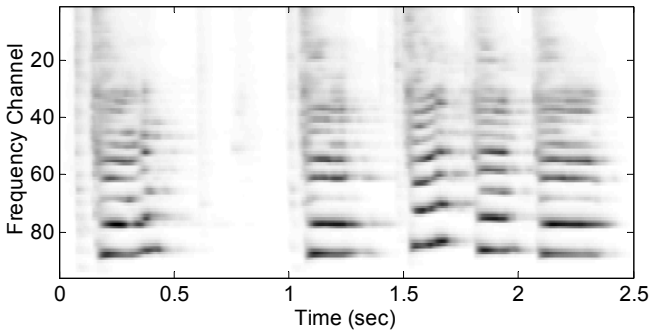


Figure 3.13. Cochleagram of a 2.5s' saxophone riff.

There, the harmonics of the sound waveform are clearly visible by the horizontal striations. Recall that higher channels correspond to lower frequencies. This picture has a limited time resolution for displaying purposes. However, the inner hair cells in the cochlea are extremely sensitive to the time structure of each component of the sound. Thus, a view of the cochleagram for a 46.44-msec time slice is presented in Figure 3.14b (on page 97). In that figure, the harmonics are not so obvious but a more precise image of auditory nerve firing responses in each channel is obtained.

As previously mentioned, cochleagrams are extremely similar to conventional spectrograms, despite their conceptual differences. In reality, both are functions of time and frequency, where pixel intensity relates to the intensity of a particular frequency component at a certain time instant. Nevertheless, the frequency axis is logarithmic in the cochleagram, as a consequence of the nearly logarithmic spacing of the center frequencies of the hair cells along the basilar membrane, while it is linear in the spectrogram. Furthermore, some onset enhancement is usually observed in cochleagrams [Slaney and Lyon, 1993]. But most importantly, cochleagrams keep the fine temporal structure of the signal in each channel.

As referred to in [Slaney and Lyon, 1993], this model is a "severe simplification" of some of the intricate operations that take place in the cochlea. The individual components in the cochlea are not accurately modeled but the overall mechanic effects are reasonably well captured, and so it does a satisfactory job in calculating a cochleagram.

3.3.2. Channel Periodicity Analysis

After determining the auditory nerve firing responses for each frequency channel, the

periodicities in the sound wave are analyzed. This is accomplished by computing the autocorrelation function in each channel, which results in a two-dimensional image of the sound signal, where the horizontal axis represents correlation lag and the vertical axis represents frequency. This image is termed *correlogram*, which literally means "picture of correlations" [Slaney and Lyon, 1993]. Each line of the correlogram contains information regarding the salience of the periodicities found for a given frequency channel. Like the cochleagram, the correlogram activity is measured by pixel intensity in the image.

The main objective of the correlogram is to summarize the temporal activity at the output of the cochlea [Slaney and Lyon, 1993]. In fact, many sounds, and particularly musical sounds, are periodic in time, or at least pseudo-periodic. The correlogram is, then, a powerful tool for detecting and visualizing the periodicities present in a signal. Hence, all channels will show peaks at the horizontal positions corresponding to correlation lags that match the periods of repetition present. Moreover, since independent ACF calculations are carried out in separate channels, the pitch detector is not affected by phase changes across channels.

Unlike the previous steps of the AMPD, the procedures for actual periodicity detection are more controversial. Indeed, whereas the former are based on direct measurements of the signal in the auditory nerve, the latter represent processing that occur in the central nervous system and is not directly observable. Namely, the use of the ACF for periodicity estimation has been a subject of criticism since some experimental studies contradict the ACF [Klapuri, pp. 28]. Anyway, the overall strategy proved successful in reproducing several phenomena in human hearing.

Slaney and Lyon argue that the correlogram is biologically plausible. In reality, a few researches suggest that the brain measures periodicities using a neural delay line, a case that is supported by the cross-correlator structures found in the brains of owls and cats (see [Slaney and Lyon, 1990]). However, there is no physiological evidence of delays that are as long as the periods of low-frequency tones [Hartmann, 1997, pp. 294].

Concerning computer implementation, the periodicities in the cochleagram are obtained by calculating the short-time ACF of the neural firing responses in each cochlear channel for a particular time window. As previously referred to, the sound wave must be divided into frames. This is equivalent to multiplying the signal by a sliding rectangular window. With the purpose of smoothing out the correlation, a Hamming window is used instead (*windowType* parameter, in Algorithm 3.1, on page 100). In order to improve efficiency, the ACF in each window is performed with the FFT algorithm, according to (3.16) (analogous to Equation (3.2) on page 68).

In (3.16), $x[n]$ represents the signal (a line in the cochleagram) as a function of sample number n in a particular time frame, $w[n]$ stands for the windowing function (a Hamming window, in this case), $x_w[n]$ is the windowed signal and $C_x[\tau]$ represents the autocorrelation of $x_w[n]$ in the corresponding time frame, as a function of lag, τ.

$$\begin{aligned} x_w[n] &= x[n] \cdot w[n] \\ C_x[\tau] &= FFT^{-1}\left|FFT\left(x_w[n]\right)\right|^2 \end{aligned} \tag{3.16}$$

It is common to normalize the ACF in order that its value at zero lag turns equal to one. However, this approach eliminates any indication of the relative power in different cochlear channels. In this way, the correlations are partially normalized by the square root of the power, so that the dynamic range of the correlogram becomes comparable to the one of the cochleagram, while still keeping the relative powers between channels [Slaney and Lyon, 1993]. The normalization is conducted as in (3.17), using the fact that the ACF at zero lag is equal to the signal power.

$$C_x[\tau] = \frac{C_x[\tau]}{\sqrt{C_x[0]}} \tag{3.17}$$

Figure 3.14b (page 97) shows a 46.44-msec correlogram frame for the saxophone riff example we have been using. This picture demonstrates the utility of correlograms in the analysis of periodic signals: vertical lines at particular autocorrelation lags can be discerned, denoting the instants when a large number of cochlear channels fire with the same period. This provides a clear indication of the pitch periods that exist in the signal.

3.3.3. Periodicity Summarization

The vertical lines across several frequency channels give evidence of pitch. Hence, a summary correlogram, or summary autocorrelation function (SACF), is computed by summing the ACFs across all channels at each time lag. This measures the likelihood that a periodicity corresponding to a particular time lag is present in the sound waveform. Moreover, the outcome of the "vertical summation" is related (although not equal) to the energy of the associated fundamental frequency. In effect, the autocorrelation value for a given F0 candidate in a channel depends on the energy of the F0 in that frequency region but it is not exactly equal to it. In addition, each channel may contain contributions from more than one note, e.g., due to harmonic collisions. Thus, the vertical summation of the autocorrelation values for each lag is only an approximation of the energy of each pitch.

As before, this way of performing periodicity summarization is also controversial. In reality, the physiological-perceptual mechanisms that combine the information in each frequency channel to infer exact pitches are complex and not completely understood yet. Therefore, until we know more precisely how the brain perceives pitch, this simple strategy seems acceptable and, probably, instructive, besides proving reasonably successful.

Finally, unlike Slaney who normalizes the summary correlogram in each frame (dividing it by the value at zero lag) [Slaney, 1998], we use its exact values in order to keep their relative saliences across frames. Such information is useful for trajectory segmentation, as will be explained in Section 4.4.

An example of a summary correlogram is presented in Figure 3.14d, where the determined pitch candidates are signaled, as will be described in the following paragraphs.

3.3.4. Salient Peak Detection

The final step of the multi-pitch detection module consists of finding a set of pitch candidates. Unlike the original Slaney and Lyon's algorithm, where only one pitch is selected in each frame (corresponding to the highest peak in the SACF), we select several pitch candidates.

A straightforward extension of standard monophonic pitch detectors is to select more than one peak in each frame. Despite the fact that this extension does not suit well the requirements for full music transcription, it worked reasonably well in the context of melody detection.

In this way, we detect the most salient peaks in the summary ACF. To accomplish this task, we first look for all peaks, i.e., local maxima, in the SACF, excluding the one at zero lag, and obtain their respective saliences, i.e., their (approximate) energies. Then, we eliminate all peaks that are not salient enough. To this end, we find the highest peak salience, *maxPeakSal* in Algorithm 3.1, and determine the minimum allowed peak salience, *minPeakSal*, using the minimum salience ratio parameter, *minSalRatio*. A maximum of 5 pitch candidates are selected in each frame (*maxNPC* parameter).

The detection of the main periodicities in our example is illustrated in Figure 3.14d, where the most salient peaks, i.e., pitch candidates, are spotted. The frequencies for the pitch candidates are then computed by inverting the periods associated with the found peaks. Finally, the pitch saliences in all frames are normalized to the [0; 100] interval, and then used in the next stages of the melody detection system.

Peak identification could be further developed by analyzing the prominence of the detected local maxima, i.e., their amplitude relatively to the neighboring valleys [Martins, 2001, pp. 31]. Hence, a minimum threshold could be defined and peaks that are not sufficiently prominent would be discarded. We evaluated this possibility but the consequence was that peaks corresponding to true pitches were often excluded. Indeed, the salience of a given peak is disturbed by the presence of other frequency components in its vicinity, which may cause significant salience reduction. Therefore, in mixtures of several sounds, the analysis of peak prominence does not seem adequate for peak detection. This was experimentally confirmed by an observed decrease in the overall pitch

detection accuracy.

3.3.5. Illustration of the Algorithm

The four steps of the AMPD algorithm are illustrated in Figure 3.14: panel a) presents a 46.44-msec frame of the saxophone riff excerpt we have been using; panels b) and c) depict the corresponding cochleagram and correlogram images, respectively; and panel d) shows the summary correlogram, where the candidate pitch periods are marked. There, it can be seen that the highest peak in the SACF (approximately at 5.4 msec, i.e., 185.2 Hz) is a multiple of the true pitch period, whose peak occurs at around 2.7 msec (i.e., 370.4 Hz). The problem of octave errors will be discussed later on.

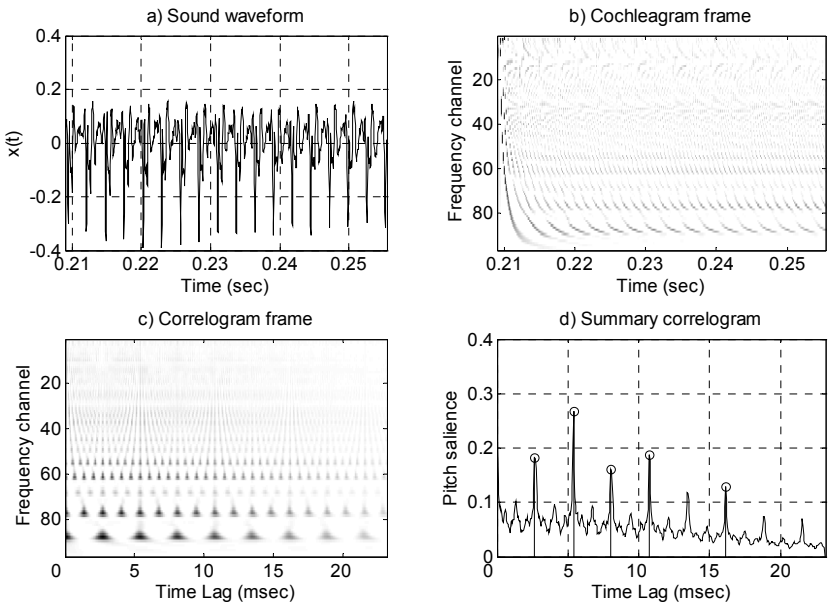


Figure 3.14. Results of the four stages of the AMPD algorithm.

3.4. Post-Processing: SACF Enhancement

As mentioned before, typical post-processor tasks are error detection and correction, smoothing, etc. In reality, regardless of the employed pitch extractor, the F0 contour is usually noisy, as well as being affected by isolated errors.

In our work, such post-processing is carried out in the later stages of the melody detection algorithm. Namely, pitch trajectories are constructed with similar F0 values, therefore disallowing pitch outliers. In addition, octave errors are dealt with during the identification of melodic notes. Hence, typical post-processing techniques such as smoothing of pitch contours, e.g., by low-pass or median filtering (see [Gómez *et al.*, 2003]) are not conducted in our system.

Instead, we perform short-term post-processing of the SACF in each frame. In fact, the SACF curve provides a good indication of the most likely periodicities in each frame of analysis but much redundant, spurious and erroneous information is also present, which makes it difficult to determine the true pitches (as seen in Figure 3.14d). Namely, peaks at multiples of the fundamental period are common.

Our strategy is based on a post-processing technique for SACF enhancement, proposed in [Tolonen and Karjalainen, 2000]. This method aims to remove much of the noisy and redundant information.

First, the SACF is expanded in time by a factor of two, as follows (3.18):

$$\begin{aligned} s_2[k] &= \begin{cases} s[k/2] & ,k \text{ even} \\ 0 & ,k \text{ odd} \end{cases} \quad ,k = 1,2,\cdots,2N \\ s_\chi[k] &= \text{interpolation}\left(s_2[k]\right) \end{aligned} \tag{3.18}$$

There, $s_2[k]$ denotes a dilated version of the original SACF, $s[k]$, whose number of samples is N. We then fill in the "empty values" of the dilated signal via linear interpolation.

Then, the enhanced SACF (ESACF), s_E[k], is obtained by subtracting the expanded SACF, $s_\chi[k]$, from the initial SACF and clipping to positive values, as in (3.19). Peak picking is then implemented on the enhanced SACF, as was described in Section 3.3.4.

$$s_E[k] = \max\left(0, s[k] - s_\chi[k]\right), \quad k = 1,2,\cdots,N \tag{3.19}$$

This procedure removes repetitive peaks with double the time lag when the basic peak is higher than the duplicate. Moreover, when the duplicate is higher, the subtraction reduces its amplitude, which may possibly become smaller than that of the basic peak. Thus, no matter if duplicates are effectively suppressed or simply attenuated, octave errors turn out to be less frequent. This is illustrated in Figure 3.15, for the saxophone example in Figure 3.14. There, it can be seen that the highest peak in the ESACF (approximately at 2.7 msec) corresponds now to the true pitch period, and so the octave error (highest peak at 5.4 msec in the top panel) is corrected. Also, the peak at the fourth multiple (at around 10.8 msec in the SACF) totally disappears in the enhanced SACF.

However, when time dilation is performed, peaks in the expanded SACF become

broader. Consequently, after subtraction, true peaks in the original SACF may become less salient or even disappear. In effect, our experimental results confirmed a reduction in the rate of octave errors, at the expense of an increased rate of false negatives. Time expansions with higher factors have even aggravated the problem.

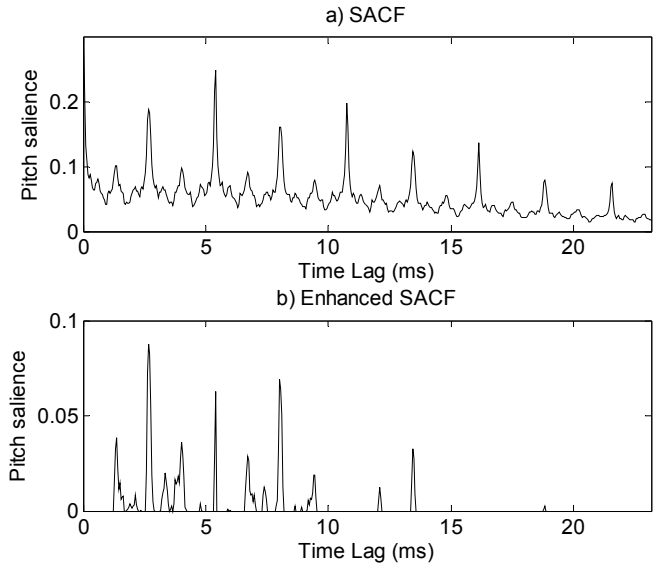


Figure 3.15. SACF Enhancement.

3.5. Putting It All Together

The complete pitch detection scheme is summarized in Algorithm 3.1. Parameter definition is presented in Table 3.1. The parameters for the cochleagram are not included, since we used the default values defined in [Slaney, 1998], as described in the text.

At this point, the motivation for extracting multiple pitches when we are only interested in the melodic line becomes clearer. Indeed, peaks corresponding to the periodicities of simultaneous notes may compete in the salience curve and be the maximum alternately. Moreover, peaks from sub or super-harmonics may also be more salient than the ones of the fundamental period, which would cause octave errors, as happens in Figure 3.14d. In reality, several pitch detection techniques, e.g., the ones based on the ACF (Section 3.1.3) are prone to octave errors.

Algorithm 3.1. Pitch detection.

0. Perform noise suppression, according to the principles of RASTA processing[45]:
   0.1. Invert the resulting spectrum to the time domain.
1. Obtain the cochleagram for each time frame:
   1.1. Apply Lyon's ear model.
2. Determine the correlogram for each time frame:
   2.1. Multiply each line of the cochleagram frame by a Hamming window (Equation (3.16)).
   2.2. Determine the autocorrelation function for each channel via FFT (Equation (3.16)).
   2.3. Normalize the ACF in each channel (Equation (3.17)).
3. Compute the summary ACF (i.e., summary correlogram) for each frame:
   3.1. Vertically sum the ACF across all channels.
   3.2. Enhance the achieved SACF[46].
4. Detect candidate peaks in the SACF.
   4.1. Detect all the peaks in the SACF.
   4.2. Determine the minimum allowed peak value (salience).
      - *maxPeakSal* ← maximum peak value.
      - *minPeakSal* ← *maxPeakSal* × *minSalRatio*.
   4.3. Eliminate the peaks with low salience:
      4.3.1. If peak salience < *minPeakSal*, eliminate peak.
   4.4. Sort the resulting peaks in descending salience order and keep the top *maxNPC* ones.
   4.5. Invert pitch periods to obtain frequencies.
5. Normalize the pitch saliences in all frames to the [0; 100] interval.
6. Return the pitch frequencies and saliences for all frames.

---

[45] As will be seen in Section 3.6, RASTA processing was excluded from the final algorithm, since it gave rise to slightly worse results. For this reason, we started the algorithm with step number zero.

[46] Again, this step was removed as a consequence of its negative effect on the final pitch detection accuracy.

Parameter Name	*Parameter Value*
frame length	46.44 msec (2048 samples, when f_s = 44100 Hz)
hop size	5.8 msec (256 samples)
RASTA frequency range	[50; 6000] Hz
windowType	Hamming
minSalRatio	0.2
maxNPC	5

Table 3.1. AMPD parameters.

Unlike some other algorithms, our method does not deal with the well-known and complex "octave problem", except for the issues described in the post-processing section. In fact, at this point it is not strictly necessary to conclude if a given pitch candidate corresponds to a real note or appears as a ghost note, whose fundamental frequency is a harmonic of some real note. Some of the ghost notes will be eliminated already at this stage based on the pitch salience threshold, whereas others will be eliminated in the following phases of the melody detection system.

Therefore, selecting several pitch candidates allows for the detection of lower-salience melody notes, which would not be captured if only a single pitch was extracted. In this way, it is possible to keep track of the global temporal continuity of each pitch. As mentioned, the motivation for this policy is that missing notes cannot be recovered afterwards but, instead, false candidates can be eliminated in later stages.

3.6. Experimental Results, Analysis and Conclusions

The evaluated pitch detectors were implemented in Matlab 7. We also made use of Malcolm Slaney's Auditory Toolbox, written in Matlab too (except for a few functions related to the ear model that were coded in C, e.g., filterbank cascade output and automatic gain control; filter design, periodicity analysis in each channel and periodicity summarization were all programmed in Matlab).

The accomplished results, which justify our selection of the AMPD, are presented and discussed in the next paragraphs. The main limitations of the proposed pitch detection scheme are discussed and suggestions for improvements are drawn.

A. Analysis of Results

We start the evaluation by comparing the performances of the analyzed pitch detectors (see Appendix A), without pre or post-processing. As referred to, at most five pitches were extracted in each frame. The performance was then evaluated by comparing the annotated F0 in each melodic frame and the closest extracted F0.

A summary of the achieved results, sorted by raw pitch detection accuracy (gray column[47]), is presented in Table 3.2. There, the average global pitch detection accuracy is presented for each of the five studied algorithms, both regarding raw (MRPA) and the chroma (MCPA) pitch accuracy. Furthermore, separate figures for our test-bed (PDB) and the MIREX'2004 database (M04) are provided. In the former, since manual annotations were conducted, the target F0s were assigned the values of the equal temperament frequencies. Hence, the extracted F0s were quantized as well to the ETFs. Due to quantization, additional pitch errors occur for notes with glissando and/or vibrato.

Pitch Detector	*Avg PDB*		*Avg M04*		*Global Average*	
	MRPA	*MCPA*	*MRPA*	*MCPA*	*MRPA*	*MCPA*
AMPD	80.2%	80.3%	81.9%	82.3%	*81.0%*	*81.2%*
Spectral ACF	69.7%	70.0%	68.5%	68.5%	*69.1%*	*69.3%*
STFT Harmonics	67.2%	67.9%	69.8%	71.9%	*68.4%*	*69.8%*
ACF	58.3%	65.0%	69.2%	72.0%	*63.5%*	*68.3%*
Probabilistic	58.1%	68.7%	55.8%	67.0%	*57.0%*	*67.9%*

Table 3.2. Comparison of pitch detection algorithms. Algorithms are sorted by raw pitch detection accuracy.

An immediate conclusion from the previous table is that the auditory-model-based pitch detector surpasses by a healthy amount all the other methods, in all evaluated metrics. Also, it can be seen that in the first three algorithms, the raw and the chroma metrics are nearly the same, i.e., harmonic-related peaks may have been extracted but the true F0 was detected as well. However, the ACF and, especially, the probabilistic pitch detector had some difficulties in this respect.

Particularly intriguing was the poor behavior of the probabilistic approach, based on [Goto, 2000]. In reality, the performance reported by the author is considerably better. As will be seen in Chapter 5, the results obtained by Tappert and Batke's system in the

[47] In the result tables presented throughout this document, we highlight specific columns by using gray shading.

MIREX'2004 evaluation (which follows rather closely Goto's system) [Gómez *et al.*, 2006] are also somewhat low. Nonetheless, the behavior of the actual PreFEst system in the MIREX'2005 evaluation was much better. Therefore, it is likely that such discrepancies derive mostly from peculiarities that neither ours nor Tappert and Batke's implementation could account for.

ID	*AMPD*		*AMPD + RASTA*		*AMPD + Enhanced SACF*		*AMPD Single Pitch*	
	MRPA	*MCPA*	*MRPA*	*MCPA*	*MRPA*	*MCPA*	*MRPA*	*MCPA*
1	97.0	97.1	96.6	96.6	93.1	93.1	28.1	61.7
2	75.8	75.9	77.9	77.9	70.1	70.6	50.1	72.0
3	89.4	89.5	88.7	88.7	83.6	83.6	77.6	82.3
4	76.4	76.7	78.8	78.8	75.5	75.5	63.6	67.4
5	62.3	62.3	64.8	64.8	54.8	54.8	34.6	49.9
6	75.2	75.2	76.9	76.8	64.5	64.6	52.1	68.6
7	95.7	95.7	96.6	96.7	88.8	88.9	77.3	91.6
8	91.7	91.7	93.8	93.8	92.3	92.3	62.5	77.1
9	55.0	55.0	61.4	61.4	48.3	48.3	46.0	46.5
10	70.9	71.2	72.7	72.7	65.7	65.7	44.1	51.6
11	92.4	92.5	88.9	88.9	91.8	91.8	37.1	85.6
12	89.7	89.7	87.5	87.5	86.4	86.5	82.1	84.3
13	93.6	93.6	92.3	92.3	92.9	92.9	60.1	78.0
14	76.5	77.4	73.9	74.1	71.0	71.0	63.6	67.7
15	81.7	82.0	79.2	79.3	80.6	80.7	54.9	67.5
16	80.8	81.7	82.3	82.4	36.2	37.4	51.7	72.7
17	73.7	73.7	73.4	73.4	74.2	74.2	60.5	67.2
18	79.7	79.9	74.9	75.0	78.9	78.9	32.2	37.2
19	75.1	75.5	65.8	66.7	61.9	62.0	25.3	31.7
20	80.3	81.1	77.1	77.9	80.1	80.5	45.3	49.1
21	88.2	88.4	86.0	86.2	87.6	87.6	64.0	67.1
Avg PDB	*80.2%*	*80.3%*	*81.5%*	*81.6%*	*75.3%*	*75.4%*	*52.1%*	*68.6%*
Avg M04	*81.9%*	*82.3%*	*79.2%*	*79.5%*	*75.0%*	*75.2%*	*54.0%*	*62.3%*
Avg	*81.0%*	*81.2%*	*80.4%*	*80.6%*	*75.2%*	*75.3%*	*53.0%*	*65.6%*

Table 3.3. AMPD results: pre and post-processing and single-pitch detection.

Based on the better behavior of the AMPD, we selected it as the pitch detector of our melody detection system. In this way, from this point forward, all the presented numbers refer to this algorithm.

After selecting the AMPD, we evaluated its accuracy when pre and post-processing was carried out. Furthermore, the case of selecting only the most salient pitch in each frame was studied. The achieved performances are presented in Table 3.3.

When looking at the first column in this table (after the ID column), we can see that, generally, the highest accuracy values (≥90%) correspond to excerpts with high SNRs (e.g., IDs 1, 3, 7, 8, 11, 12 and 13 - see description of excerpts in Appendix B). The exception is in the opera samples (IDs 18 and 19), which we suppose to be a consequence of very fast pitch variations in some notes with extreme vibrato (both in amplitude and frequency). As will be seen in Figure 3.16 (page 106), the employed pitch detector responds reasonably well to fast pitch variations. However, in extremely fast situations, it looks like the assumed frame stationarity no longer applies, making pitch detection more vulnerable (the SACF becomes noticeably noisier).

In the same column, we observe that excerpts with medium/high SNR have accuracy values around 80%. For samples where the SNR is not so favorable, the performance drops to values close to 70% and would have decreased even more if excerpts from styles such as dance music had been used. As expected (and confirmed by the obtained results) in those cases many melodic pitches pass undetected, on account of being masked by more intense sounds that stem mostly from strong percussive sounds.

The particularly low performance of two excerpts in the first column, namely Ricky Martin (5) and Eliades Ochoa (9), caught our attention. In both cases, glissando was often present in the attack of the notes, which has given rise to errors in the computation of pitch accuracy, not on its detection (recall that a fixed frequency value was assigned to all the time frames of a target note, due to manual annotation). Also, especially in Eliades Ochoa's excerpt, it was found that the used tuning did not match the ETFs. Owing to quantization, semitone errors have led to this low performance. Nonetheless, both problems will be tackled in the next chapter, where equal temperament notes are identified, coping with note dynamics as well as the tuning issue.

Concerning percussive "noise", no benefits arose from the use of RASTA pre-processing, as can been in Table 3.3. Indeed, the attained pitch detection accuracy was nearly the same with and without noise suppression. In some excerpts, a few originally less salient true peaks become now more evident, but the improvements are not significant. This is not surprising since the underlying RASTA assumption that the harmonic peaks are clearly above the noise floor means that the harmonic peaks can be easily detected. Another effect of RASTA processing was that the filtered signals sounded "blurred". After the conducted analysis, we decided not to include the pre-processing stage, since the overall pitch detection accuracy decreased slightly.

As for SACF enhancement, the accomplished figures were usually below the ones achieved without post-processing. This was particularly notorious in a MIDI synthesized sample (16), where they decreased dramatically. In reality, the subtraction of the SACF may induce a noteworthy attenuation, or even to the disappearance, of several true peaks. This seems to have occurred thoroughly in that MIDI excerpt.

Finally, in order to quantitatively justify our strategy of selecting more than one pitch per frame, we evaluated the AMPD when only one pitch was selected in each frame (the most salient one in the SACF). As expected, the performance has worsened substantially. However, the discrepancy between raw and chroma pitch accuracy directed our attention to a few interesting cases. In fact, in some situations the inaccuracy of the single-pitch detector was either as a consequence of octave errors (from real pitches or harmonic peaks) or just because the melodic pitch was less intense than the pitch of other simultaneous notes. The first situation corresponds to the following excerpts: Battlefield Band (ID 11, instrument playing one octave below), Hallelujah (2, chords from the choral ensemble), Claudio Roditi (7, instrument one octave below) or midi1 (16, detection of sub or super-harmonics). In these samples, the chroma accuracy is approximately the same as the one attained when multiple pitches are selected. In some other excerpts, the pitch accuracy diminished since the melodic pitch was simply less intense than the pitch from other simultaneous notes. This is the case of Pachelbel (ID 1, strong bass), Avril Lavigne (6, strong guitars with distortion) or daisy3 (13, strong guitars). Finally, in some excerpts both situations occurred, namely in Pachelbel (1, strong bass notes one octave below) or Ricky Martin (5, strong bass, sometimes octave-related).

Regarding the behavior of the opera excerpts in the single-pitch evaluation, the low values in the MRPA metric are not surprising because of octave errors. Even so, a much better performance was expected in the MCPA measure. While analyzing the obtained results, we noticed that the frequency of the highest peak in the SACF was often "almost" harmonically-related to the annotated frequency. To illustrate, one of the analysis frames had a target F0 of 625 Hz and a pitch candidate at 630 Hz (a difference of 13.8 cents), but the highest peak had an associated frequency of 165.8 Hz, while the theoretical 4^{th} sub-harmonic was expected at 156.3 Hz (a difference close to one semitone).

In terms of dynamics, the described pitch detection algorithm can cope reasonably well with fast pitch variations. This is illustrated in Figure 3.16 for an opera excerpt (opera female in Table 2.1). There, the continuous curve denotes the annotated F0, whereas the dots represent the extracted F0 sequence. Fine precision in tracking strong vibrato conditions (across 3 semitones with around 7 Hz vibrato frequency, in this example) is observed. Nonetheless, it is important to say that in extremely fast pitch variations such exactness is not so pronounced, since the precise location of the relevant peaks in the summary correlogram becomes harder to resolve due to the lack of signal stationarity. Anyway, from a melodic transcription point of view, small pitch differences are acceptable, as long as the actual MIDI note number is correctly determined. However, in appli-

cations such as expressiveness analysis, precise pitch information is crucial.

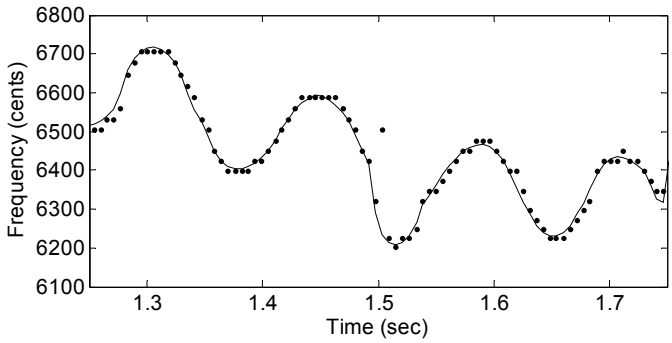


Figure 3.16. Response of the AMPD to fast pitch variations.

B. Limitations of the Algorithm and Possible Improvements

The main shortcoming of the AMPD is its high computational cost. In fact, in our computing system, the algorithm takes an average of 24 min and 24 sec to complete the analysis of a 20 sec's excerpt, which corresponds to about 97% of the total computational time of melody detection. This is a result of the high number of filters used and of the almost entire native Matlab implementation. Porting the code to C could reduce the execution time by an order of magnitude. Restricting the filterbank to a narrower frequency range would also reduce the execution time. In effect, by default the frequency range of the filterbank goes up to the Nyquist frequency; anyway, this range could be narrowed, since the amplitudes of the higher harmonics in musical instruments are usually low. A typical maximum frequency of 5 kHz should suffice, given the finer sensitivity of the human ear in that range and the characteristics of musical instruments. A related but simplified version proposed in [Tolonen and Karjalainen, 2000] and based on [Meddis and Hewitt, 1991], where only two channels are used, could be exploited as well. The authors argue that, despite the model simplifications, the method qualitatively retains the performance of multi-channel systems. However, they did not conduct any real tests using continuous complex song excerpts.

As for peak masking in the SACF, it is likely that the AMPD could be improved by carrying out frequency analysis in each channel, in order to compensate for noisy frequency regions where masking is more likely to occur. But this would increase even more this already costly algorithm.

The followed pitch detection approach does not accomplish voicing analysis, i.e., it assumes that an F0 is always present, except in frames where the signal energy is exces-

sively low. This is sometimes untrue, e.g., in noisy or percussive segments. Nevertheless, solely by analyzing a temporal frame, we cannot assure if it corresponds to an isolated percussive sound or a mixture of different, possibly harmonic, sounds. Therefore, we simply select the highest peaks in each frame's SACF, which are then either used or ignored if reasonable peak continuation can be accomplished or not. Hence, ghost tracks usually emerge, as will be seen in the next chapter.

In our peak detection scheme, we did not add any constraints as to the closeness of adjacent peaks. This is particularly evident in noisy SACFs, leading sometimes to the selection of peaks with very close frequency values. As a consequence, some ambiguities in peak continuation may arise, as we will discuss in Chapter 4 (fortunately, these are not very frequent). Moreover, this strategy is not very faithful to the related physiologic mechanisms in the ear. Indeed, an important phenomenon of auditory physiology is lateral inhibition (critical-band masking), in which the presence of a strong peak makes the ear less sensitive to sounds of about the same frequency [Hartmann, 1997, pp. 256]. In a first attempt to deal with this issue, low-pass filtering was utilized. However, the net effect was that the overall pitch detection accuracy decreased because true peaks with low prominence were filtered out. We could improve this, for instance by implementing peak validation using the detected pitches in the past few frames, where matched pitches would be preferred over close unmatched ones.

In relation to the suggested memory-based approach, the algorithm could also be improved by taking advantage of the evidence that abrupt F0 changes are not very common in musical signals (except for extreme vibrato conditions). Thus, the pitches found in previous frame(s) could support the detection of F0 candidates in a given analysis frame. In this way, low salience peaks (that would normally be excluded) could now be selected based on their occurrence in recent frames.

Despite the fact that our pitch detection scheme performed reasonably well in a melody extraction task (at least in the used dataset), the employed peak detection methodology may produce both ghost pitch candidates (e.g., sub or super-harmonics, noisy peaks) and false negatives. Regarding particularly the *maxNPC* parameter, we found experimentally that lower values give rise to a high number of missing melodic pitches, whereas higher values jeopardize the selection of the final melody, since too many notes are created (as will be discussed in Chapter 5). Therefore, it would be interesting to check if this stage could be improved by making use of techniques from polyphonic pitch detection, as the ones described at the beginning of this chapter.

Furthermore, computing the SACF just by vertical summation in the correlogram might be problematic since several sounds occur simultaneously, interfering with each other. In reality, sounds exhibiting inharmonicities or mixtures with harmonic collisions cause incorrect pitch saliences.

Inharmonicities originate deviations from the ideal harmonic structure and so the

periodicities detected in the higher channels do not exactly match the periodicities in the lower channels. For example, in piano sounds a slight left shift in periodicity may be observed in the upper channels because the frequencies of higher harmonics are slightly above their theoretical values. Consequently, the obtained peak may become broader, have lower amplitude than it should or, most likely, deviate a little from the frequency of the first harmonic, usually considered as the F0. Nonetheless, with respect to the latter point, the achieved pitch is more accurate in perceptual terms, as pointed out in [Slaney and Lyon, 1990].

As for harmonic collisions, no procedure was executed to split apart the energies of common harmonics that come from different sources. Hence, after vertical summation, the pitch salience for a given time lag contains both the energy of a note at that pitch and the energy of common harmonics from notes at different F0s.

Inharmonicity and, most importantly, harmonic collisions could be tackled, at least to some extent, by extending the devised periodicity summarization method. Namely, we could pursue a multi-pitch detection approach such as Klapuri's [Klapuri, 2003], where spectral smoothness and iterative estimation and cancellation are conducted. Moreover, in another work, Klapuri implemented a few modifications to an auditory-model-based pitch detector developed by himself and Jaakko Astola [Klapuri and Astola, 2002; Klapuri, 2004, pp. 44] that could possibly be applied to the auditory model we adopted. There, the ACF calculations are replaced by a technique termed harmonic selection and a more complex sub-band-weighting is performed for the combination of results across bands. Basically, the salience of each F0 candidate is computed using only their respective partials and ignoring the spectrum between the partials. Furthermore, multiple F0 candidates are determined following an intricate iterative estimation and cancellation strategy: the predominant pitch is first selected and cancelled; the estimation is then repeated for the residual sound and the next detected pitch removed; this is repeated for the number of sounds present.

It is also important to refer that periodicity computation based on the ACF may induce a large peak corresponding to the root tone of a chord. This represents a false pitch candidate and is a limitation in music analysis, namely for full music transcription (e.g., detection of note chords is difficult to carry out in this manner). However, with respect to melody detection, a leading musical part is assumed to exist (see Section 2.3). In this way, a clear peak at the fundamental period (or at a multiple) is usually found. Additionally, even when root tone chords appear, these might be discarded in the subsequent stages of the melody detection system in case their pitches are sufficiently low.

C. Other Possible Improvements

Available content and context information might be used as well for supervising the pitch detection process. For example, if we knew beforehand the instrument used for

sound production, algorithmic parameters could be adapted and optimized accordingly. Moreover, different pre-processing methods might suit better particular instruments (e.g., singing voice pre-processing requirements are necessarily different from the ones of wind instruments; in string instruments, their specific resonant properties might need to be dealt with; also, inharmonicity might have to be considered in instruments such as piano). Additionally, available information on the key or melodic profile could guide the F0 detection process. Kashino and colleagues made use of content and context information with recourse to internal sound source models [Kashino *et al.*, 1995]. Since we focus on an entirely automatic system for melody detection, if key or instrument information were used, these should be automatically "learned" by the algorithm.

Finally, in the implemented ear model the envelope is not extracted in each band, unlike in [Meddis and O'Mard, 1997; Meddis and Hewitt, 1991]. Thus, it would be interesting to evaluate the behavior of the used model if a low-pass filter was applied after half-wave rectification, i.e., keeping mostly spectral interval information.

Chapter 4

FROM PITCHES TO NOTES

> *"The messages in language are built out of a limited set of units (i.e., phonemes). Similarly, the messages in music are built out of a limited set of units (i.e., scaled notes)"*
>
> *Stephen Handel, "Listening: An Introduction to the Perception of Auditory Events", 1989 (pp. 332)*

Several applications of melody detection, namely melody transcription, query-by-melody or motivic analysis, require the explicit identification of musical notes, which allow for the extraction of higher-level features that are musicologically more meaningful than the ones obtained from low-level pitches.

Despite the importance of the note as the basic representational symbol in Western music notation, the explicit and accurate recognition of musical notes is somewhat overlooked in automatic music transcription research. In effect, most approaches disregard the importance of notes as musicological units having dynamic nature.

Therefore, in this chapter we propose a mechanism for quantizing the temporal sequences of the detected F0s into note symbols, characterized by precise temporal boundaries and note pitches (namely, MIDI note numbers). The developed method aims to cope with typical dynamics and performing styles such as vibrato, glissando or legato.

Section 4.1. Introduction

We start this chapter with an analysis of the importance of the note in Western music. We then review the existing strategies for explicit note determination.

Section 4.2. Pitch Trajectory Construction (PTC)

Our scheme for detection of musical notes starts with the construction of a number of pitch tracks, formed by connecting consecutive pitch candidates with similar fre-

quency values. The objective is to find regions of stable pitches, which indicate the presence of musical notes.

Here, the frequencies in each track are first quantized to MIDI note numbers. Then, peak continuation based on frequency proximity is accomplished, allowing track inactivity and tackling possible ambiguities via a look-ahead procedure. Short tracks are then eliminated and the unused pitch candidates are reassigned to the validated tracks.

Section 4.3. Frequency-Based Track Segmentation

Since the obtained trajectories may contain more than one note, temporal segmentation must be carried out. This is performed in two phases, recurring to the pitch and salience contours of each track, i.e., frequency and salience-based segmentation.

In frequency-based segmentation, the objective is to separate all notes of different pitches that are included in the same trajectory, coping with glissando, legato and vibrato, as well as other sorts of frequency modulation. Moreover, the precise timings of each note candidate are adjusted.

After segmentation, a final MIDI note number is assigned to each note candidate. The devised note labeling approach deals with possible tuning inaccuracies.

Section 4.4. Salience-Based Track Segmentation

With respect to salience-based segmentation, the objective is to separate consecutive notes at the same pitch that may have been incorrectly interpreted as forming one single note. To this end, the note candidates that arise from the previous step are segmented based on pitch salience minima, which mark the temporal boundaries of each note.

To increase the robustness of the algorithm, note onsets are detected directly on the audio signal and used to validate the salience minima found in each note candidate.

Section 4.5. Putting It All Together

The complete note determination scheme is summarized in algorithmic form and model parameters are listed in this section.

Section 4.6. Experimental Results, Analysis and Conclusions

Finally, experimental results are presented and examined. The main pros and cons of the followed methodology are analyzed and directions for future improvements are pointed out.

4.1. Introduction

In this section, we discuss the importance of the note as a basic representational symbol in Western music and recycle the discussion on tonal fusion and perception of musical notes. Then, we review the main strategies towards note detection, both in monophonic and polyphonic contexts.

4.1.1. The Note as a Basic Representational Symbol

As mentioned, the note is the fundamental building block of Western music notation. When characterizing a musical note (for example in a written score), features such as *pitch*, *intensity*, *rhythm* (typically representing accents and timing information, e.g., duration, onset and ending time), *performance dynamics* (glissando, legato, vibrato, tremolo, etc.) and sometimes even *timbre* are considered. Hence, in this respect, the goal of any automatic transcription system would be to capture all this information.

We explicitly extract pitch, intensity and timing data. As for note dynamics, this is implicitly conveyed in the pitch and intensity contour of each note; however, we will not explicitly state, e.g., if a not has vibrato or tremolo, though the analysis of the contours could provide such information. Concerning note timbre, we conducted some efforts to model it, with the purpose of implementing note clustering in the next chapter.

While the note is central in Western music notation, it is not evident if the same applies when we talk about perception. In reality, some researchers defend that, instead of notes, humans extract auditory cues that are then grouped into *percepts*, i.e., brain images of the acoustical elements present in a sound. Eric Scheirer argues that "most stages of music perception have nothing to do with notes for most listeners" [Scheirer, 2000, pp. 69]. In fact, he adds, "the acoustic signal must always be considered the fundamental basis of music perception", since "[it] is a much better starting point than a notation invented to serve an entirely different mode of thought" [Scheirer, 2000, pp. 68].

Namely, tonally fused sounds seem to play an important role in music perception [Scheirer, 2000, pp. 30]. For example, as referred to in Section 2.4, the sounds produced by pipe organs perceptually fuse into one single percept, i.e., the various concurrent sounds are unconsciously perceived as a whole. Thus, trying to explicitly extract the individual musical notes that are enclosed in a tonally fused sonic object seems perceptually unnatural.

Nevertheless, we could also argue that notes are indeed perceived in some situations, for instance while listening to monophonic melodies or to songs where the melody obeys our previous definition. In such cases, the average listener easily memorizes them and replicates what he hears, for example by humming or whistling. In addition, he can even try to mimic the timbre of the singer, as well as some of the performance dynamics. In

other words, his mental constructs seem to correspond to musical notes, although he may or may not be aware of that.

Regardless of the arguments that can be presented to support or reject the note as a perceptual construct, the identification of musical notes is essential in music transcription, in order for a symbolic representation to be derived. As a result, in our work we consider musical notes as the basic building blocks of music transcription and, therefore, investigate mechanisms to efficient and accurately identify them in musical ensembles.

4.1.2. Current Approaches for Note Determination

The identification of musical notes is one of the less explored areas in the field of automatic music transcription. Regarding the particular melody transcription problem, this is confirmed by the absence of a note-oriented metrics in the audio melody extraction track of MIREX'2005.

Past work in the field addressed especially the extraction of pitch lines, without explicit determination of notes, or using ad hoc algorithms for the segmentation of pitch tracks into notes (e.g., segment as soon as MIDI note numbers change). This has turn out to be difficult for some signals, particularly for singing [Klapuri, 2004, pp. 3]. In effect, the presence of glissando, legato, vibrato or tremolo makes it sometimes a challenging task, as illustrated in Figure 3.16. Yet, amplitude and frequency modulation are important aspects to consider when segmenting notes.

Different kinds of methodologies for note determination, e.g., note segmentation and labeling, are summarized in the following paragraphs.

A. Note Segmentation

Amplitude-based Segmentation

In monophonic contexts, note segmentation is typically accomplished directly on the temporal signal. In fact, since no simultaneous notes occur, several systems first implement signal segmentation and then assign a pitch to each of the obtained segments, e.g., [Chai, 2001, pp. 48]. In this strategy, silence detection is frequently exploited, as this is a good indicator of note beginnings and endings. In algorithmic terms, silences correspond to time regions where the amplitude of the signal (the root mean square energy is generally used) falls bellow a given threshold. The robustness of these methods is usually improved by employing adaptive thresholds [McNab *et al.,* 1996b; Chai, 2001].

Other related, yet more elaborate schemes, tackling especially the transcription of the singing voice in monophonic audio, are pursued in [Haus and Pollastri, 2001; Clarisse *et al.,* 2002], where several procedures are carried out in order to improve the

reliability of the determined segments. For instance, Goffredo Haus and Emanuele Pollastri measure the RMS power of the signal and compare it with a signal-to-noise threshold for segment boundary detection. In each segment voiced regions are then identified, which are the ones used for pitch detection.

The main limitations of amplitude-based segmentation come from the difficulties in accurately defining amplitude thresholds (particularly in polyphonic contexts, where sources interfere severely with each other). This may give rise to both excessive and missing segmentation points, namely to the unsuccessful separation of notes played legato. Moreover, in a polyphonic context several notes may occur at the same time, with various overlapping patterns. Consequently, note segmentation cannot be performed neither before nor independently of pitch detection and tracking.

Frequency-based Segmentation

Frequency variations are usually better indicators of note boundaries, especially in polyphonic contexts. Here, frame-wise pitch detection is first conducted and then pitch changes between consecutive frames are used to segment notes. To this end, frequency proximity thresholds are normally employed, e.g., [McNab *et al.*, 1996b].

However, several of the developed systems do not adequately handle note dynamics. This is frequently the case in transcription systems dedicated to specific instruments such as piano, which do not modulate substantially in pitch, e.g., [Hawley, 1993].

In [Martins, 2001], pitch trajectories are created with recourse to a maximum frequency distance of half a semitone. Nevertheless, smooth frequency transitions between notes might lead to trajectories with more than one note. This was not attended to apparently because most of the used excerpts came from MIDI-synthesized instruments played without note legato.

Martin bases the identification of musical notes on the continuation of pitches across frames and on the detection of onsets. This information is combined and analyzed in a blackboard framework [Martin, 1996]. The used frequency proximity criteria used are not described but, apparently, note hypotheses may contain more than a single note in the case of smooth pitch transitions. The provided examples are not conclusive since tests were implemented with piano sounds only, characterized by having sharp onsets and not modulating significantly in frequency.

The problem of trajectories containing notes of different pitches was addressed in [Eggink and Brown, 2004]. There, the frequency distance is computed based on an average of the past few F0 values. The authors argue that this allows for vibrato while breaking up successive tones even when they are separated by only a small interval. However, even in this situation, it is not guaranteed that individual tracks will contain one single note. Indeed, depending on the defined threshold, smooth frequency transitions between consecutive notes could still be kept in a single track, as we have experimentally

confirmed. In this situation, the frequency values in the transition may not differ considerably from the average of the previous values. In other situations, the two notes could be segmented somewhere during the transition, rather than at its beginning. Also, the use of a small interval is not robust to missing pitches in tracks containing vibrato, which could generate abrupt frequency jumps.

In brief, the main drawback of the previous methodologies is that the balance between over and under-segmentation is often difficult: if small frequency intervals are defined, the frequency variations in fast glissando or vibrato zones might be erroneously separated into several notes; on the other hand, if larger intervals are permitted, a single segment may contain more than one note.

Probabilistic Frameworks for Frequency-based Segmentation

Some of the weaknesses described above are tackled under probabilistic frameworks. Namely, Timo Viitaniemi and colleagues employ a probabilistic model for converting pitch tracks from monophonic singing excerpts into a discrete musical notation (i.e., a MIDI stream) [Viitaniemi *et al.*, 2003]. The used pitch-trajectory model is an HMM whose states correspond to MIDI note numbers, where an acoustic database is utilized to estimate the observation probability distribution. In addition, a musicological model estimates the key signature from the obtained pitch track, which is used to give information on the probability of note occurrence. Finally, inter-state transition probabilities are estimated based on a folk song database and a durational model is used to adjust state self-transition probabilities according to the tempo of the song (known a priori). The output of the HMM is the most likely sequence of discrete note numbers, which (ideally) copes with both pitch and performing errors. Note boundaries then directly denote transitions of MIDI numbers. Moreover, note durations are adjusted recurring to tempo information.

Ryynänen and Klapuri handle note segmentation in the context of a polyphonic transcription system [Ryynänen and Klapuri, 2005a]. The overall strategy is very elegant and apparently robust. There, two probabilistic models are used: a note event model, used to represent note candidates, and a musicological model, which controls the transitions between note candidates by using key estimation and computing the likelihoods of note sequences. In the note event model, a three-state HMM is allocated to each MIDI note number in each frame. The states in the model represent the temporal regions of note events, comprising namely an attack, a sustain and a noise state, and therefore taking into consideration the dynamic properties and peculiarities of musical performances. State observation likelihoods are determined with recourse to features such as the pitch difference between the measured F0 and the nominal pitch of the modeled note, pitch salience and onset strength. The observation likelihood distributions are modeled with a four-component GMM and the HMM parameters are calculated using the Baum-Welch algorithm. The note and the musicological models then constitute a probabilistic note

network, which is used for the transcription of melodies by finding the most probable path through it using a token-passing algorithm. Tokens emitted out of a note model represent note boundaries.

Segmentation of Consecutive Notes at the Same Pitch

In the systems where segmentation is primarily based on frequency variations, consecutive notes with equal pitches are often left unsplit. This occurs both when legato is performed and when a maximum inactive time (normally referred to as "sleeping time") is allowed in pitch tracking. This track inactivity is often tolerated in order to handle situations when pitches pass undetected in a few frames, despite the fact that the respective note is sounding.

Approaches that do not permit track inactivity or admit it only during very short intervals usually cause over-segmentation. This seems to be the case of Bello's method (described in [Gómez *et al.*, 2006]). Although not many details are provided, we can presume that the creation of pitch tracks did not allow sufficient frame inactivity, since a profusion of fragments corresponding to the same note often results.

In [Eggink and Brown, 2004], frame sleeping is consented to and notes are then split when abrupt discontinuities in F0 intensity occur. However, this simple scheme suffers from the same shortcomings associated with amplitude-based note segmentation, namely regarding the accurate definition of thresholds: a satisfactory balance between over and under-segmentation is hard to attain.

This problem is partly solved in [Kashino *et al.*, 1995], where terminal point candidates, which correspond to clear minima in the energy contour of each pitch track, are either validated or rejected according to their likelihood and on the detected rhythmic beats. This is much more robust than using only amplitude information but, even so, consecutive notes occurring in between beats may be left unsegmented.

In the note segmentation scheme described in [Ryynänen and Klapuri, 2005a], it this not obvious how this issue is addressed. In reality, the connections between the three states in the models of note events are not strictly left-to-right: the attack state has a left-to-right connection with the sustain state, but this and the noise state might alternate. Thus, when a token is sent to the attack of another note event, a segmentation boundary becomes evident, no matter whether the MIDI note number is the same or not. However, when there is a transition from the noise to the sustain state in a note model, it is not clear if pitch was undetermined for a while or if two consecutive notes at the same pitch were present.

B. Note Labeling

After segmentation, a note label has to be assigned to each of the identified seg-

ments. Typically, pitch detection is executed on short time frames and the average F0 in a segment is quantized to the frequency of the closest equal temperament note, e.g., [McNab *et al.,* 1996a; Martins, 2001]. This averaging strategy might deal well with frequency modulation, but does not seem appropriate when glissando is present.

In other approaches, the average F0 is computed in the central part of the note, since pitch errors are more likely to occur at the attack and at the decay [Clarisse *et al.,* 2002]. In monophonic transcription systems, filtering may be implemented as well to cope with outliers or octave errors [Clarisse *et al.,* 2002]. In addition, the median of F0 values may be used rather than the average.

C. Adaptation to Instrument and Singer's Tuning

Methodologies for note labeling should handle the case where songs are performed off-key, e.g., when the instruments are not tuned to the equal temperament frequencies. This is also frequent in monophonic singing, since only a few people have absolute pitch. Also, non-professional singers (no matter if they have absolute pitch or not) have a tendency to change their tuning during longer melodies, typically downwards, as referred to in [Ryynänen, 2004, pp. 27].

Some systems attend to this problem, particularly in the transcription of the singing voice or in the adaptation of note labeling to the intonation of the performer, e.g., [McNab *et al.,* 1996b; Haus and Pollastri, 2001; Viitaniemi *et al.,* 2003; Ryynänen, 2004]. Namely, Rodger McNab and colleagues [McNab *et al.,* 1996b] devise a scheme for adjusting note labeling to the own tuning of individual users. There, a constantly changing offset is employed, which is initially estimated by the difference between the sung tone and the nearest one in the equal temperament scale. Then, the resulting customized musical scale continuously alters the reference tuning, in conformity with the information from the previous note. This is based on the assumption that singing errors tend to accumulate over time. On the other hand, Haus and Pollastri [Haus and Pollastri, 2001] assume constant sized errors. There, note labeling is achieved by estimating the difference from a reference scale (the equal temperament scale in this case), then conducting scale adjustment and finally applying local refinement rules.

The described approaches make sense in monophonic contexts, where we readily know that all the obtained notes represent the melody. Then, individual singer tuning can be estimated using the set of sung notes. But the same does not apply in polyphonic contexts, where notes from different parts are simultaneously present. In this case, slight departures from the equal temperament scale may occur in singing[48]. However, since many notes are present and note clustering is a complex task to accomplish (as will be seen in Chapter 5), it is difficult to estimate the tuning of a particular singer (or instru-

[48] This occurs, for example, in a few notes of an excerpt from Eliades Ochoa (Table 2.1).

ment). Therefore, we propose a different heuristic for dealing with deviations from the equal temperament scale, which is partly based on the assumptions that off-key instrumental tuning is not significant, and neither are tuning variations in singing, as the employed songs are performed by professional singers in a stable instrumental set-up.

4.2. Pitch Trajectory Construction (PTC)

In the identification of the notes present in a musical signal, we start by creating a set of pitch trajectories, formed by connecting pitch candidates with similar frequency values in consecutive frames. The idea is to find regions of stable pitches, which indicate the presence of musical notes. In order not to lose information on note dynamics, e.g., glissando, legato, vibrato or tremolo, we took special care to ensure that such behaviors were kept within a single track. The PTC algorithm receives as input a set of pitch candidates, characterized by their frequencies and saliences, and outputs a set of pitch trajectories, which constitute the basis of the future musical notes.

In perceptual terms, such pitch trajectories correspond, to some extent, to the perceptually atomic elements referred to in Section 2.2.2. In effect, in the earlier stages of sound organization, the human auditory system looks for sonic elements that are stable in frequency and energy over some time interval [Ellis, 1992, pp. 30]. In our work, we only resort to frequency information in the development of these atoms. Anyway, energy information could have also been incorporated for the sake of perceptual fidelity. Actually, we have exploited it to disentangle situations of peak competition among different tracks, but frequency information proved sufficient even in such cases.

We follow rather closely Xavier Serra's peak continuation mechanism ([Serra, 1989, pp. 61-70; Serra, 1997]). Nonetheless, since we have a limited set of pitch candidates per frame, our implementation becomes lighter. In fact, Serra looks for regions of stable sinusoids in the signal's spectrum, which leads to a trajectory for each found harmonic component. In this way, a high number of trajectories have to be processed, which makes the algorithm a bit heavier, though the basic idea is the same. Moreover, as the number of pitches in each frame is small, these are clearly spaced most of the time, and so the ambiguities in trajectory construction are minimum.

The main tasks carried out are described in the next paragraphs. This procedure is grounded on three main parameters: a maximum frequency difference between consecutive frames, a maximum inactivity time in each track and a minimum trajectory duration.

4.2.1. MIDI Quantization

In the first step, we quantize the collected F0 candidates to their closest MIDI notes.

This method, which differs from continuation based on exact frequency values in the original algorithm, was conducted because we have experimentally found that peak continuation utilizing MIDI note numbers promotes a more robust trajectory build up, as will be seen. Furthermore, the representation of notes using MIDI numbers simplifies an eventual representation of the sound waveform in MIDI format (e.g., for generation of a MIDI file). Nevertheless, the initial frequency values are still kept, since they contain important information for the analysis of note dynamics.

The conversion from frequency values to MIDI note numbers, f_{MIDI}, is executed in this manner (4.1):

$$f_{MIDI}[k] = \text{round}\left(\frac{\log\left(f[k]/F_{ref}\right)}{\log \sqrt[12]{2}}\right),\ F_{ref} \approx 8.1758Hz \tag{4.1}$$

where $f_{MIDI}[k]$ represents the MIDI note number associated with frequency f in the k^{th} frame and F_{ref} is the reference frequency, which corresponds to MIDI number zero.

4.2.2. Peak Continuation based on Frequency Proximity

The trajectory creation scheme relies on three parameters, as mentioned. The first parameter, *maxSTDist*, represents the maximum frequency distance in semitones for continuing trajectories. In order not to lose information on note dynamics, we took special attention to guarantee that such features were kept within a single track. In addition, it is important to cope with frequency oscillations that might have resulted from noise and interference from other sources. As an example, undetected pitches (dealt with by allowing a maximum inactivity time, as will be described) could give rise to abrupt frequency jumps in situations of high-frequency and high-amplitude vibrato, and consequent erroneous note segmentations. Also, situations of fast glissando, with long frequency jumps in consecutive frames, would not be kept within one single track.

To this end, we defined *maxSTDist* as one semitone since the amount of frequency changing in vibrato, for both the singing voice and musical instruments, is typically around one semitone[49] [Handel, 1989, pp. 177]. In practice, separations of almost 2 semitones are permitted, due the fact that continuation uses MIDI numbers. Exemplifying, in a trajectory whose last note is MIDI 70, and having a continuation candidate with note number 71, if the respective original frequency values correspond to the lower limit of note 70 and to the upper limit of note 71, a difference close to two semitones results.

[49] Naturally, this value may vary significantly. For example, the vibrato of lyric singers may reach much broader pitch variations (e.g., three semitones in the opera excerpt illustrated in Figure 3.16). As for the frequency of vibrato, typical values are close to 6 Hz [Handel, 1989, pp. 177].

This proved to be more robust than calculating the distance employing the initial frequencies.

In this way, the described dynamical features are satisfactorily kept within a common track, instead of being separated into a number of different trajectories, e.g., one trajectory for each note that a glissando may traverse. Hence, a single trajectory may contain more than one note and, therefore, trajectory segmentation based on frequency variations is carried out in the next stage of the melody detection algorithm.

To be more precise, even if a low frequency distance were imposed, some trajectories could contain more than one note, because of smooth transitions between notes, e.g., in legato performances. To cope with this situation, some authors (e.g., [Eggink and Brown, 2004]) compare the maximum allowed distance to the frequency average of the last few frames. However, as discussed in Section 4.1.2, it is not assured that individual tracks will contain only one note. Also, this strategy is not robust to missing pitches in tracks with vibrato, which could cause abrupt frequency jumps.

4.2.3. Track Inactivity

One important aspect to consider in any pitch tracking methodology is that pitches might pass undetected in some frames as a result of noise, masking from other sources or low peak amplitude.

For this reason, permitting a trajectory to "sleep" for a while and "wake up" when its pitch reappears can be regarded as simple implementation of the Gestalt principle of continuity, described in Section 2.2.2. To illustrate, if a note played by a flute is masked by a loud event, such as a drum, which occurs more or less in the middle of the note, the auditory system will typically "hear through" the drum sound and assume that the note was there when the drum was hit, in spite of not explicitly detecting it. This is true as long as the approximate frequency content of the note is kept.

Thus, the second parameter, *maxSleepLen*, specifies the maximum time where a trajectory can be inactive, i.e., when no continuation peaks are found. If this number is exceeded, the trajectory is stopped. A maximum of 62.5 msec was defined (corresponding to 11 frames, which, in practice leads to 63.9 msec). For inactive frames, both the frequency and salience values are set to zero. As a result, many sparse trajectories arise (most of them relating to weak notes), which might still be part of the melody.

Though this value may seem too high, it was intentionally selected. Indeed, lower maximum inactivity times usually give rise to a profusion of short trajectories at the same MIDI number. This is due to the fact that, in polyphonic signals, pitch masking occurs more notoriously than in monophonic audio. Therefore, these should be merged later on. On the contrary, admitting a longer maximum inactivity time has the drawback that

consecutive notes at the same pitch may be kept within only one track. To this end, trajectory segmentation, now based on salience variations, must be performed.

The reason why we prefer the "track splitting" over the "track merging" paradigm is that, even with a perfect pitch detector, consecutive notes at the same pitch might be integrated into one single track, e.g., when notes are played legato. The energy decreases but no silence actually occurs and so track splitting had to be conducted anyway.

4.2.4. Tackling Ambiguities

In general, the pitches selected in each frame are clearly spaced, as a consequence of accepting only a small number of them in the AMPD. Hence, the peak continuation procedure is usually unequivocal. However, some ambiguities may occur mostly in situations where close peaks are selected. In reality, owing to the allowance of a wider maximum frequency difference, close frequency peaks may compete and induce ambiguous continuations that might possibly end up in trajectory construction errors.

In this way, we extended the algorithm by introducing a look-ahead scheme to prevent trajectories from continuing pitch candidates that would give rise to erroneous continuations in future frames.

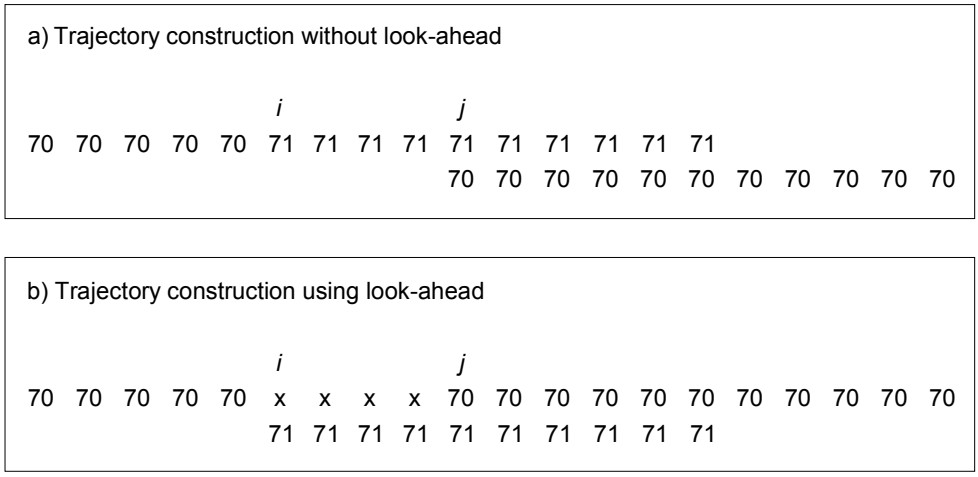


Figure 4.1. Look-ahead procedure.

Exemplifying, imagine that the last MIDI note number of a given track is 70 and that we had continued it with a pitch candidate with number 71, at frame *i* (Figure 4.1a). Then, a few frames ahead (i.e., less than *maxSleepLen* frames), in frame *j*, we had found both notes 70 and 71. In this situation, that trajectory would have erroneously contin-

ued note number 71 instead of note 70. Indeed, since the same MIDI number is present ahead, it would have been wiser not to continue the candidate note 71 and sleep for a while until the correct note was found. Note 71 would then have been used to continue or create another trajectory. Therefore, when noisy peaks are detected in the neighborhood of true peaks, a more reliably assignment of pitch candidates to tracks usually results. This is illustrated in Figure 4.1b.

4.2.5. Elimination of Short Tracks

The last parameter, *minTrajLen*, controls the minimum trajectory duration. Here, all finished tracks that are shorter then this threshold, defined as 125 msec (22 frames, leading in reality to 127.7 msec), are eliminated. This parameter was set in conformity with the typical note durations in Western music. As Bregman points out, "Western music tends to have notes that are rarely shorter than 150 msec in duration. Those that form melodic themes fall in the range of 150 to 900 msec. Notes shorter than this tend to stay close to their neighbors in frequency and are used to create a sort of ornamental effect" [Bregman, 1990, pp. 462].

4.2.6. Reassignment of Unused Pitch Candidates

Another extension of the original peak continuation algorithm consisted of reducing pitch track sparseness. In fact, due to the allowed sleeping time, trajectories may have a variable number of empty frames.

To this end, pitch candidates in rejected tracks are used to fill in the blanks in other trajectories. For example, in Figure 4.1, if the trajectory with MIDI note number 71 was eliminated, its values from frame i to frame j-1 would be used to fill in the empty frames of the other track, giving that the restriction for maximum semitone difference was fulfilled. Furthermore, the deleted track might had been used to either anticipate the beginning or postpone the end of the other trajectory, as long as the maximum semitone distance was satisfied. As a result, phenomena such as frequency drifting at the attack and decay regions of musical notes are kept within a single track, instead of being split apart. This in turn promotes more accurate note timings.

4.2.7. Putting It All Together

This algorithm is graphically illustrated in Figure 4.2 (adapted from [Martins, 2001, pp. 43]). There, black squares represent the candidate pitches in the current frame n. Black circles connected by thin continuous lines indicate trajectories that have not been fin-

ished yet. Dashed lines denote peak continuation through sleeping frames. Black circles connected by bold lines stand for validated trajectories, whereas white circles represent eliminated trajectories. Finally, gray boxes indicate the maximum allowed frequency distance for peak continuation in the corresponding frame.

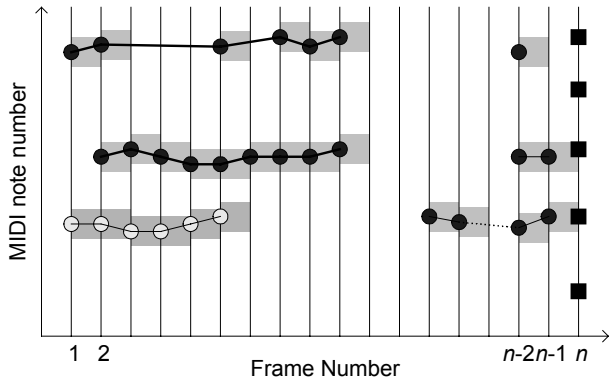


Figure 4.2. PTC algorithm.

Algorithm 4.1 summarizes the operations carried out for pitch trajectory construction. Parameter definition is presented in Table 4.1.

The results of the process for the simple saxophone riff example we used before are presented in Figure 4.3 (page 126). There, we can see that some of the obtained trajectories comprise glissando regions. Also, some of the trajectories include more than one note and should, thus, be segmented.

Algorithm 4.1. Pitch trajectory construction.

1. Quantize frequencies to the closest MIDI note numbers (but keep original frequency values – Equation (4.1)).
2. Create initial trajectories:
   2.1. Use MIDI note numbers, frequencies and saliences of the pitch candidates in the first non-empty frame.
3. Perform peak continuation, i.e., for all frames:
   3.1. Get the note numbers of all pitch candidates in the current frame.
   3.2. Define all the continuation possibilities for each

non-finished track, i.e., note numbers in the current frame that are within the *maxSTDist* range.

3.3. Assign new continuation note numbers to each non-finished track, i.e., for all non-finished tracks:

3.3.1. Select the best MIDI candidate for continuation, i.e., the closest one that does not have any other closer trajectories:

- If the selected note number is different from the last one, handle continuation ambiguities via the look-ahead procedure (Figure 4.1).
- In case of tie, compare exact frequency distances rather than MIDI number differences.

3.3.2. If the trajectory is continued:

a) Update the trajectory length.

b) Add the current note number, frequency and salience to it.

3.3.3. Otherwise:

a) Increment the number of inactive frames, *numSleep*.

b) If *numSleep* $\geq$ *maxSleepLen*, stop the current trajectory.

3.4. Eliminate or validate the stopped trajectories, i.e., for all stopped trajectories:

3.4.1. If length < *minTrajLen*:

a) Eliminate the current trajectory.

b) Use its pitch candidates to fill in the empty frames in one non-finished trajectory (the first one found whose MIDI note numbers are in the *maxSTDist* range) or to pad its content to the onset and/or ending of the found track.

3.5. Create new tracks, i.e., for all non-continued pitch candidates:

3.5.1. Create a new trajectory, initialized with the present note number, frequency and salience.

4. Return all finished trajectories (in ascending start frame order).

Parameter Name	*Parameter Value*
maxSTDist	1 semitone
maxSleepLen	62.5 msec (11 frames → 63.9 msec)
minTrajLen	125 msec (22 frames → 127.7 msec)

Table 4.1. PTC parameters.

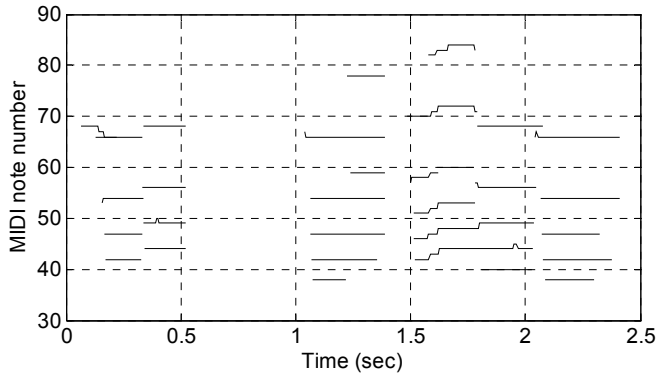


Figure 4.3. Results of the PTC algorithm.

4.3. Frequency-Based Track Segmentation

The trajectories that result from the PTC algorithm may contain more than one note and, therefore, must be divided in time. In frequency-based track segmentation, the goal is to split notes of different pitches that may be present in the same trajectory, coping with glissando, legato, vibrato and other sorts of frequency modulation.

4.3.1. Note Segmentation

The main issue with frequency-based segmentation is to approximate the frequency curve by piecewise-constant functions (PCFs), as a basis for the definition of MIDI notes. However, this is often a complex task, since musical notes, besides containing regions of nearly stable frequency, also comprise regions of transition, where frequency evolves until (pseudo-)stability, e.g., glissando. Additionally, frequency modulation may also occur,

where no stable frequency exists. Yet, an average stable fundamental frequency can be determined.

Our problem could thus be characterized as one of finding a set of piecewise-constant/linear functions that best fits to the original frequency curve, under the constraint that it encloses the F0s of musical notes. As unknown variables, we have the number of functions, their respective parameters (slope and bias - null slope if PCFs are used), and start and endpoints.

We have investigated some methodologies for piecewise-linear function approximation. Two main paradigms are defined: "characteristic points" (CPs) and "minimum error" (ME). Algorithms based on CPs do not suit well our needs, e.g., in the case of frequency modulation, and so we constrained the analysis to the ME paradigm. This one can be further categorized into two main classes [Pérez and Vidal, 1992]. In the first one, an upper bound for the global error is specified and the minimum number of functions that satisfies it, and respective parameters, is computed. This situation poses some difficulties, mostly associated with the definition of the maximum allowed error. In effect, an inadequate definition may lead to a profusion of PCFs in regions of vibrato. In the second (less studied) class, a maximum number of functions is specified, and optimization is conducted with the objective of minimizing the global fitting error. However, these approaches either require that an analytic expression of the curve be known, or need to test different values for the number of functions. Hence, methods in this class do not seem to suit our needs either.

In this way, we propose an approach for the approximation of frequency curves by PCFs, taking advantage of some peculiarities of musical signals.

A. Filtering of the Original Frequency Curve

The algorithm starts by filtering the frequency curves of all tracks, in order to fill in missing frequency values that result from the PTC stage. This is carried out by a simple zero-order-hold (ZOH), as in (4.2). There, $f[k]$ is the frequency value in the current track for its k^{th} frame and $f_F[k]$ denotes the filtered curve.

$$\forall_{k\in\{1,2,\cdots,N\}}, f_F[k] = \begin{cases} f[k], & \text{if } f[k] \neq 0 \\ f_F[k-1], & \text{if } f[k] = 0 \end{cases} \tag{4.2}$$

B. Definition of Initial PCFs

Next, the filtered frequency curve is approximated by PCFs through the quantization of each frequency value to the corresponding MIDI note number, according to (4.1). Therefore, PCFs can be directly defined as sequences of constant MIDI numbers, as in (4.3). There, PC_i represents the i^{th} PCF, defined in the domain D_i and characterized by a

sequence of constant MIDI numbers equal to c_i. Also, the particular case of singleton domains is considered. The total number of PCFs is denoted by *nPC*.

$$\begin{aligned} &\forall_{i\in\{1,\cdots,nPC\}}, \\ &1: \quad D_i = \{a_i,\cdots,b_i\} = \{k \in \{1,2,\cdots,N\} : f_{MIDI}[k] = c_i\} \\ &2: \quad PC_i[k] = c_i \quad , \forall_{k\in D_i} \end{aligned} \tag{4.3}$$

C. Filtering of PCFs

However, because of frequency variations resulting from modulation or jitter, as well as frequency errors from the pitch detection stage, fluctuations of MIDI note numbers may occur. Also, glissando transitions are not properly kept within one single function. Consequently, $f_{MIDI}[k]$ must be filtered so as to allow for a more robust determination of PCFs that may represent actual musical notes. Four stages of filtering are applied with the purpose of dealing with short PCFs, i.e., PCFs whose length is below *minNoteLen* (equal to *minTrajLen*, i.e., 22 frames = 127.7 msec), where the length of a function is the number of elements in its domain.

The initial filtering stages recur to the presence of long PCFs (having lengths above *minNoteLen*), which provide good hints for function merging.

Oscillation Filtering

For this reason, in the first filtering stage, sequences of PCFs with alternating values are detected and merged (i.e., sequences of PCFs with MIDI note numbers *c* and *c*+1, or *c*+1 and *c*). Such oscillations can be combined in a more robust way in case they are delimited by long PCFs. The general methodology proceeds like this:

1. We start by looking for a long PCF;
2. Next, we search for functions with alternating MIDI numbers until another long PCF is found again;
3. The detected oscillations indicate regions of frequency modulation and, therefore, the respective PCFs are fused as follows:
   a) If the delimiting functions have the same MIDI number, that the resulting PCF receives this value;
   b) On the other hand, if the last function has a different MIDI number, it is not obvious which pitch should be assigned. Hence, we sum the durations of the short PCFs in between for each of the two possible MIDI note numbers and select the winner as the most frequent one. In order to account for empty frames in the pitch track under analysis, only non-empty frames are used when counting the occurrences of each MIDI note number;

c) The alternating short PCFs are then combined with the corresponding initial long PCF.

This procedure is illustrated in Figure 4.4, where the thick lines denote long PCFs and thin ones represent short functions.

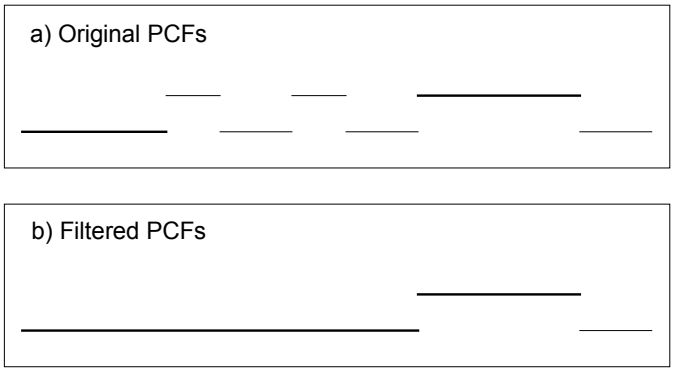


Figure 4.4. Oscillation filtering.

Filtering of Delimited Sequences

In the second stage, the goal is to combine short PCFs that are delimited by two PCFs with the same note number (one of them must be long). This may occur due to pitch jitter from noise, pitch detection errors or tuning issues. Such "enclosed" functions are handled in this fashion:

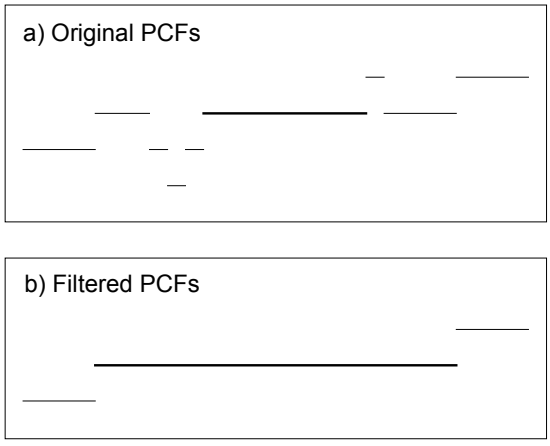


Figure 4.5. Filtering of delimited sequences.

1. Once again, we start by looking for a long PCF;
2. Then, we search forward for another PCF with the same MIDI number;
3. If the sum of the durations of all the PCFs in between is short, those functions and the delimiting ones are merged;
4. We then repeat from step 2, but now to the left of the long PCF found.

This is exemplified in Figure 4.5.

Glissando Filtering

Next, sequences representing glissando are analyzed as described below (and illustrated in Figure 4.6):

1. As before, we first look for a long PCF;
2. After that, we search for a succession of short PCFs with constantly increasing or decreasing MIDI numbers (corresponding to the transition region) and possibly ending with a long PCF;
3. The detected transition region suggests a possible glissando, treated as follows:
   a) If the final PCF in the sequence is long, the merged PCF maintains its value, based on the evidence that the glissando evolved until the long function;
   b) Otherwise, if the sequence contains only short PCFs and if the duration of the whole sequence is long enough to form a note, the fused PCF receives the value of the most frequent MIDI note number (the last PCF may result from frequency drifting at the ending, and so it does not obtain preference).

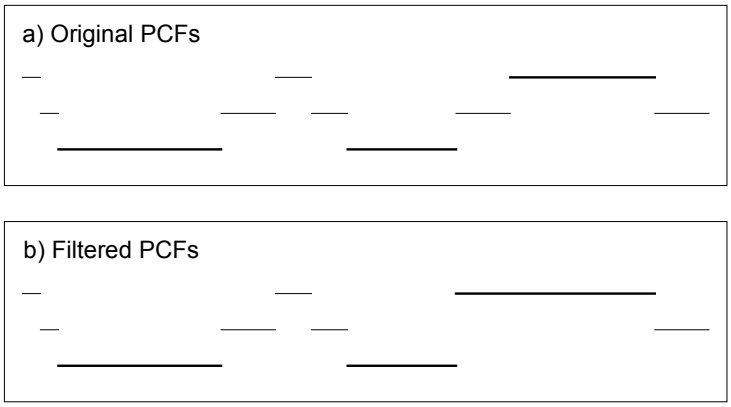


Figure 4.6. Glissando filtering.

Filtering without the Requirement of Finding Long PCFs

After making use of long PCFs for filtering, a few short PCFs may still be present, as can be seen in Figure 4.6. Therefore, two final stages of filtering are applied, much in the same way as filtering of glissando and of delimited sequences was performed, with the difference that no long PCFs need to be found.

In this way, filtering of delimited sequences is first conducted, where we search for a short PCF and then for another PCF after it with equal note number, complying with the procedure described at the top of page 130. Step 1 is executed differently, since short PCFs are now looked for.

As for glissando filtering, we look for sequences indicating glissando transitions (as defined on page 130) starting with short PCFs, and proceeding like this:

1. If the final PCF in the sequence is long, the new PCF keeps its value, as before;
2. Otherwise, if the sequence is long enough to form a note, the new PCF receives the value of the most frequent MIDI note number, also as before;
3. Otherwise, the last MIDI number may correspond to frequency drifting at the decay region. Thus, the sequence of PCFs is merged with the immediately precedent long PCF.

Final short note filtering is illustrated in Figure 4.7.

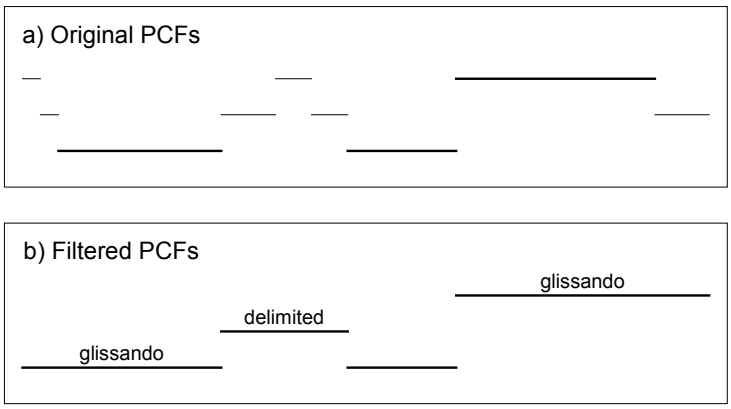


Figure 4.7. Final short note filtering.

D. Time Adjustment

After filtering, the precise timings of each PCF must be adjusted. Indeed, as a consequence of MIDI quantization, the exact moment where transitions start is often de-

layed, since the frequencies at the beginning of transitions may be converted into the previous MIDI number, instead of to the next MIDI number.

Hence, we define the start of the transition as the point of maximum derivative of $f[k]$ after it starts to move towards the following note, i.e., the point of maximum derivative after the last occurrence of the median value.

The median, md_i, is calculated only for non-empty frames (zero frequency) whose original MIDI note numbers are kept after filtering, according to (4.4). In this way, the median is obtained in a more robust way, since possibly noisy frames are not considered.

$$md_i = \text{median}\left(f\left[k\right]\right) \quad , \forall_{k \in D_i : f_{MIDI}[k]=c_i \text{ and } f[k] \neq 0} \tag{4.4}$$

The discrete derivative is computed using the filtered frequency curve, as in (4.5):

$$\dot{f}\left[k\right] = f_F\left[k\right] - f_F\left[k-1\right] \tag{4.5}$$

4.3.2. Note Labeling

Once pitch tracks are segmented into regions of different pitch, we have to assign a final MIDI note number to each of the defined PCFs.

Accurate note labeling of singing voice excerpts is usually not trivial because of the enriched dynamics added by many singers. Moreover, human performances are often unstable (e.g., tuning variations) and affected by errors (e.g., pitch singing errors). These difficulties are not so severe in our circumstances, since we employ recordings of professional singers in stable instrumental set-ups. Therefore, we assume that singing tuning variations are minimum and that the instrumental tuning does not depart significantly from the reference equal temperament scale.

In order to increase the robustness of the assignment procedure, we deal with ambiguous situations where it is not obvious which the correct MIDI number should be. This happens, for instance, when the median frequency is close to the frequency border of two MIDI notes, as in recordings where tuning variations in singing occur (e.g., our Eliades Ochoa's excerpt) or when instruments are tuned off-key.

A. Definition of the Initial MIDI Note Number and the Allowed Frequency Range

Thus, we determine the initial MIDI note number from the median frequency, md_i, of each function, according to (4.1). Then, we calculate the ETF associated with the obtained MIDI number, by inverting (4.1). This is carried out with the purpose of checking if the median does not deviate excessively from the reference frequency. Here, we define

a maximum distance, *maxCentsDist*, of 30 cents, as in (4.6):

$$\begin{aligned} iniMIDI_i &= \mathrm{MIDI}\left(md_i\right) \\ refF_i &= \mathrm{frequency}\left(iniMIDI_i\right) \\ range_i &= \left[refF_i \cdot 2^{-\max CentsDist/1200}; refF_i \cdot 2^{\max CentsDist/1200} \right] \end{aligned} \tag{4.6}$$

There, $iniMIDI_i$ represents the candidate MIDI number of the i^{th} PCF, $refF_i$ stands for the corresponding ETF, $range_i$ denotes the allowed frequency range and 'frequency' is a function for figuring out the ETF from a MIDI note number (i.e., inversion of the 'MIDI' function, defined in (4.1), disregarding the rounding operator).

B. Determination of the Final MIDI Note Number: Tuning Compensation

Consequently, if the median is in the permitted frequency range of the respective MIDI number, there is evidence that the assigned MIDI number is correct, and so we keep it.

However, when the median deviates significantly from the reference, it is not clear whether the initial MIDI number is correct or not. In order to clarify this ambiguity, we use a simple heuristic for the determination of the final MIDI number. Basically, if the median is higher than the upper range limit, the final MIDI number may need to be incremented[50]. This is conducted using the following scheme:

1. We first calculate the frequency value in the frontier of the two candidate MIDI numbers, $borderF_i$, which is 50 cents above the reference frequency of the initial MIDI note number, (4.7):

$$borderF_i = refF_i \cdot 2^{50/1200} \tag{4.7}$$

2. Next, we count i) the number of frames, *numH*, for which the frequency is above the frontier, i.e., the number of frequency values corresponding to the incremented MIDI number and ii) the number of frames, *numL*, where the frequency is below the median. Then:
   a) If *numH* > *numL*, we conclude that the final MIDI number should be changed to the incremented value;
   b) Otherwise, it is left unchanged.

[50] We describe the analysis carried out using as example the upper range. In any case, we proceed likewise if the median is below the lower range limit, except that in this case the note number might need to be decremented.

4.3.3. Merging of Simultaneous PCFs with Equal MIDI Note Numbers

Finally, in case PCFs that are simultaneous in time receive the same MIDI note number, these are merged together into one single PCF, beginning with the PCF that starts earliest and ending with the last one to finish.

In order to assign frequency and pitch salience values to the frames in the common time intervals, we select, in each frame, the frequency (and respective pitch salience) that is closest to the ETF of the obtained MIDI note number.

Anyway, this situation rarely happens since, as referred to in Section 4.2.4, the trajectory construction scheme does not entail many ambiguities due to the relative distance between the detected peaks.

4.3.4. Putting It All Together

The description of frequency-based segmentation is condensed in Algorithm 4.2. Parameter definition is presented in Table 4.2.

Algorithm 4.2. Frequency-based track segmentation.

1. Apply ZOH to the original frequency curve, f (Equation (4.2)).
2. Define initial PCFs:
   - 2.1. Quantize the filtered curve, f_F, to the corresponding MIDI note number (Equation (4.3)).
3. Filter the set of initial PCFs:
   - 3.1. Perform oscillation filtering, based on the detection of long PCFs (Figure 4.4).
   - 3.2. Implement filtering of delimited sequences, resorting to the detection of long PCFs (Figure 4.5).
   - 3.3. Perform glissando filtering, based on the detection of long PCFs (Figure 4.6).
   - 3.4. Execute the same kind of filtering without the requirement of finding long PCFs (Figure 4.7):
     - 3.4.1. Implement filtering of delimited sequences.
     - 3.4.2. Perform glissando filtering.
4. Adjust the timings for each PCF, according to the point of maximum derivative of the frequency curve.

5. Assign a MIDI note number to each PCF:
   5.1. Compute the initial MIDI number as the one corresponding to the median frequency md_i (Equation (4.4))
   5.4. Determine the final MIDI note number, taking into consideration tuning compensation.
6. Merge PCFs that are simultaneous in time and have received the same MIDI note number.
7. Return the resulting notes.

Parameter Name	*Parameter Value*
minNoteLen	125 ms (22 frames → 127.7 msec)
maxCentsDist	30

Table 4.2. Parameters for frequency-based track segmentation.

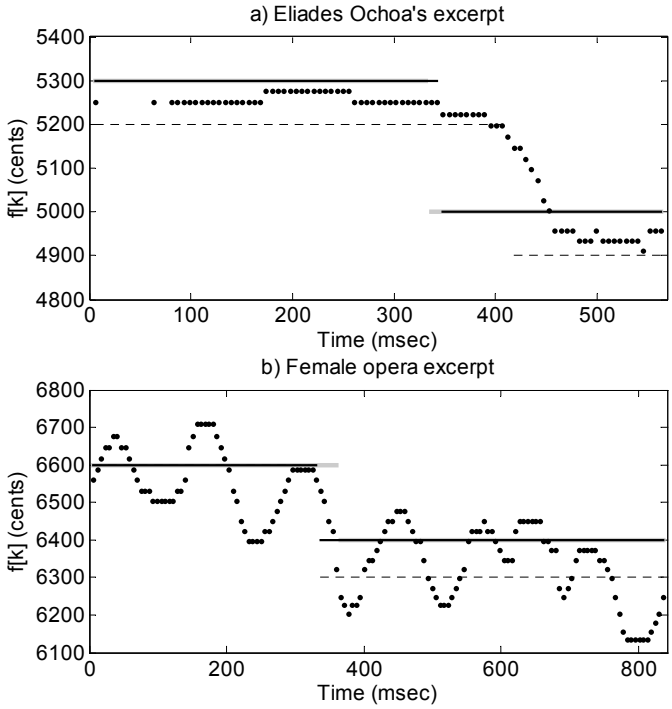


Figure 4.8. Results of the frequency-based track segmentation algorithm.

The results of this algorithm are illustrated in Figure 4.8, for a pitch track from Eliades Ochoa's "Chan Chan" and the female opera excerpt presented in Table 2.1. There, dots denote the F0 sequence under analysis, gray lines are the reference segmentations, dashed lines denote the results attained prior to time correction and final note labeling and solid lines stand for the final achieved results. It can be seen that the segmentation methodology works quite well in these examples, despite some minor timing errors that may have even derived from annotation inaccuracies. The results for the sketched opera track, where strong vibrato is present, are particularly satisfactory.

4.4. Salience-Based Track Segmentation

As for salience-based track segmentation, the objective is to separate consecutive notes at the same pitch, which the PTC algorithm may have interpreted as forming one single note. This requires segmentation based on pitch salience minima, which mark the limits of each note. In fact, the salience value depends on the evidence of pitch, which is correlated (though not exactly equal) to the energy of the F0 under consideration. Consequently, the envelope of the salience sequence is similar to an amplitude envelope: it grows at note attack, has then a more steady region and decreases at the decay. Thus, notes can be segmented by detecting clear minima in the pitch salience sequence[51].

4.4.1. Candidate Segmentation Points

In a first attempt towards salience-based segmentation, we developed a prominent valley detection method for determining salient minima corresponding to candidate segmentation points.

A. Filtering of the Original Salience Curve

As in the frequency-based segmentation stage, we start by filtering the salience sequence with a ZOH, due to missing values.

B. Looking for Clear Salience Minima

After ZOH filtering, we iteratively look for all clear local minima of the filtered salience sequence, $s_F[k]$, i.e., sufficiently prominent minima (as defined below). This is car-

[51] A pitch salience sequence is formed by the succession of pitch saliences in the frames of a given pitch track. This should not be confused with the pitch salience curve defined in Section 3.3.3 as a summary ACF for pitch detection.

ried out in this manner:

1. First, all local minima and maxima are found, coping with plateaus. In these situations, the indexes of the corresponding minima/maxima are assigned to the middle of the plateau;
2. Then, only deep enough minima are selected. This is accomplished in the following recursive procedure:
   a) The global minimum of $s_F[k]$ is found;
   b) Next, the set of all local maxima is divided into two subsets, one to the left and another one to the right of the global minimum;
   c) The global maximum for each subset is then determined;
3. The global minimum is selected as a clear minima if its prominence, i.e., the difference between its amplitude and that of both the left and right global maxima, is above the defined minimum peak-valley distance, *minPvd*;
4. Finally, the set of all local minima is also divided into two new intervals, to the left and to the right of the global minimum.
5. The described operations are then recursively repeated for each of the new subsets until all deep minima and respective prominences are found.

C. Remarks on the Detection of Candidate Segmentation Points

One difficulty of the proposed scheme is its lack of robustness. In effect, the best value for *minPvd* was found to vary from track to track, along different song excerpts. Indeed, a unique value for that parameter leads to both missing and extra segmentation points. Also, it is sometimes difficult to distinguish between note endings and amplitude modulation in some performances.

Therefore, we improved our method by implementing note onset detection directly on the audio signal and matching the obtained onsets with the candidate segmentation points that resulted from the detection of prominent valleys.

In this way, *minPvd* should receive a low value so that missing segmentation points are unlikely. In addition, this parameter ought to be adaptive in order to cope with differences in salience ranges across different notes. Hence, *minPvd* was set to 10% of the maximum amplitude range of the salience sequence under consideration (whose values belong to the [0; 100], after the normalization conducted in the pitch detection stage). Due to the defined low value, extra false segmentation points occur, which are eliminated later on via onset matching. Moreover, as a consequence of employing a low threshold, the encountered minima are not that "clear" any longer.

Figure 4.9 illustrates the algorithm for detection of candidate segmentation points. There, the pitch salience sequence of a trajectory from Claudio Roditi's performance of

"Rua Dona Margarida" is depicted, where 'o' represent correct segmentation candidates and '*' denote extra segmentation points.

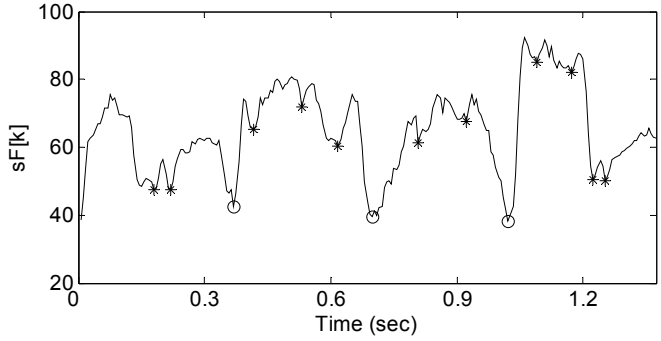


Figure 4.9. Results of salience-based track segmentation: initial candidate points.

4.4.2. Onset Detection

The objective of onset detectors is to accurately locate the beginnings of musical notes in acoustic signals. Humans perceive onsets as a consequence of noticeable changes in the intensity, pitch or timbre of a sound [Klapuri, 1999].

Robust onset detection is a demanding task, even for monophonic recordings. For example, most methodologies that rely on variations of the amplitude envelope behave satisfactorily for sounds with sharp attacks, e.g., percussion or plucked guitar strings, but show some difficulties when notes are played with little amplitude inflection, for example, in glissando or in intentionally smooth attacks, e.g., bowed violin strings. The problem is all the more acute in polyphonic mixtures, where energy bursts from the attacks of different notes overlap in time.

A substantial amount of research related to onset detection has been recently conducted. This constitutes a research topic of its own and for this reason a detailed study of the subject is out of the scope of our work. On the contrary, we used the pragmatic strategy of basing our efforts on state-of-the-art techniques. For an overview of the area see, e.g., [Bello *et al.*, 2005; Klapuri, 1999].

The algorithm implemented in our system is largely based on [Klapuri, 1999], with some adaptations. His approach shows some similarities with parts of Eric Scheirer's mechanism for tempo and beat analysis [Scheirer, 1998], from which we have also borrowed a few elements.

A. Band-Pass Filtering

The central idea is to perform onset detection in a band-wise manner. A bank of nearly critical band filters is chosen, covering the frequencies from 44 Hz to the Nyquist frequency. For a sampling frequency f_s = 22050 Hz, 20 filters result, where the first one is low-pass, the last one is high-pass and the remaining are band-pass.

Elliptic filters are employed, in order to ensure a maximally sharp cutoff in the transition band, as proposed in [Scheirer, 1998]. Since it is important to maintain the temporal properties of the signal in each band, we imposed zero-phase as a requirement. Thus, bi-directional filtering [Smith, 1997, pp. 330] is carried out as in (4.8):

$$S_B^i(z) = S(z) * H^i(z) * H^i\left(z^{-1}\right) \tag{4.8}$$

There, $S_B^i(z)$ represents the filtered output at band i, $H^i(z)$ denotes the filter discrete transfer function at the same band and $S(z)$ represents the original signal, all in the Z-domain. As a consequence of bi-directional filtering, and according to the desired final transfer function, we specified the following filter parameters: 3rd order filters, with 1.5 dB ripple in the pass-band and 20 dB of rejection in the stop-band. The design parameters are (roughly) doubled because of bi-directional filtering, e.g., 6th order filters result, in conformity with the definitions in [Scheirer, 1998]. As for the cutoff frequencies, the three lowest filters are one-octave wide BPFs, whereas the remaining are third-octave wide BPFs, with no band overlapping.

B. Determination of Energy Variations in Each Band

After filtering, the objective is to compute the energy variations in each band, as a basis for the detection of onset components. To this end, the amplitude envelope in each band is first extracted via rectify-and-smooth. This is accomplished like this:

1. The output of each band is first decimated to 200 Hz, so as to make calculations easier;
2. Then, the decimated outputs of each band are full-wave rectified and smoothed with a 100 msec half-Hanning window, as in (4.9):

$$\begin{aligned} s_H^i[k] &= \left|s_B^i[k]\right| * w\left[k - {}^{W}\!/_{2}\right] \\ w[n] &= 0.5 - 0.5\cos\left(\frac{2\pi(n-1)}{W-1}\right),\ n = 1,2,\cdots,{}^{W}\!/_{2} \end{aligned} \tag{4.9}$$

There, $w[n]$ denotes the half-Hanning window and $s_H^i[k]$ represents the smoothed output at band i. This window approximates reasonably well the energy integration performed by the human auditory system [Scheirer, 1998].

Again, we guarantee zero output delay by shifting the window;

3. Next, information about energy variations in each band is achieved by computing the first-order derivatives, $\dot{s}_H^i[k]$, of the amplitude envelopes, $s_H^i[k]$.
4. Finally, the derivative curve in each band is half-wave rectified, since we are only interested in the points of positive energy variations.

C. Integration of the Information in All Bands and Onset Detection

The collected information is then integrated across all bands and then onsets are detected in this fashion:

1. First, we linearly sum the calculated derivatives and look for noticeable maxima in the summed derivative. This is executed much in the same way as in the previous procedure for detection of clear minima, except that now we search for peaks instead of valleys;
2. The summed derivative curve is then normalized to the [0; 1] interval;
3. After that, we select initial onset candidates by finding all peaks whose saliences are above *minPeakSal* = 0.05;
4. We then delete components that are closer than *minOnsetDiff* = 50 msec to a more intense component, since some peak neighborhoods may be very dense, [Klapuri, 1999];
5. Finally, clear onsets, i.e., onsets with amplitudes above *clearOnsetMag* = 0.4, are selected.

D. Illustration and Remarks on the Onset Detection Approach

The results of onset detection are illustrated in Figure 4.10, for Claudio Roditi's excerpt. There, the onset salience curve is depicted and the identified onsets are circled. Dashed vertical lines denote the manually annotated onset times of the melody notes. As can be seen, there is almost a perfect match between the annotated and the obtained onset times. However, several false negatives appear, e.g., around 1.62 and 1.87 sec, since the respective peak amplitudes are excessively low. Nevertheless, even in those situations the annotated onsets match very well the observed peaks.

The previous example was particularly simple, since there are only two simultaneous voices plus percussion. Moreover, the two parts have synchronous onset times. The main difficulty comes from a few notes played legato, which give rise to a decrease in onset sharpness and its consequent undetection. In more complicated examples, several false positives result, which correspond to the onset times of notes from other voices apart from the melody, as well as from percussion beats and noise.

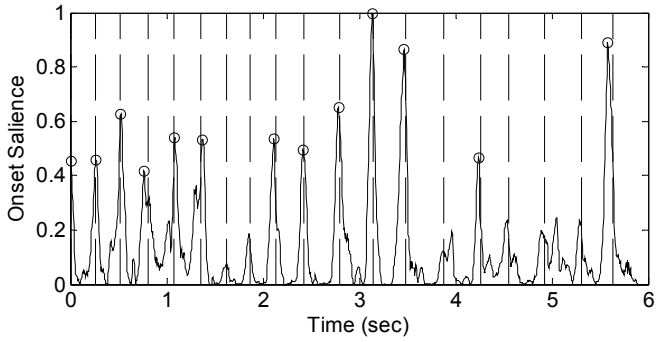


Figure 4.10. Results of onset detection.

In [Klapuri, 1999], a relative derivative function (RDF) was used instead of the first-order derivative (FOD). The RDF is computed by dividing the FOD by the amplitude envelope. The idea is to deal with the fact that some sounds, especially the ones with slow attacks, may take some time to reach the point where the amplitude is maximally rising. In those situations, the measured onset time is delayed, in comparison to the physical one. Also, the amplitude during note attack is not strictly monotonically increasing, which would lead to several local maxima in the first-order derivative. However, after experimental testing, some problems were encountered. Namely, in our saxophone riff, the determined onsets occurred ahead of time, as depicted in Figure 4.11.

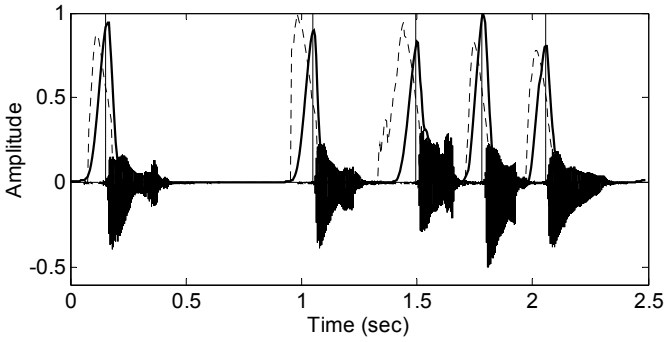


Figure 4.11. Comparison of onset results using the RDF and FOD functions.

In the previous figure, the original waveform is represented by the curve with negative values, the dashed line stands for the onset salience curve from the RDF, the solid line denotes the onset salience curve based on the FOD and the straight vertical lines denote the acquired onset times from the FOD. There, it can be seen that the peaks of the onset salience curve obtained from the RDF occur much too early. This seems to be

a side-effect of breathing noise before the attacks of the notes. We evaluated the two approaches in our database and best results were achieved with the first-order derivative.

Another difference from the method in [Klapuri, 1999] is that, there, algorithm onset components were detected in each band before integration, whereas in our implementation, integration is performed before the detection of onset components. No substantial accuracy differences were observed and early integration was lighter in computational terms.

Also, the threshold for onset definition was originally psychoacoustically-inspired. Our implementation is less robust in this respect, but it is simpler.

4.4.3. Validation of Candidate Segmentation Points

After onset detection, our goal is to validate the previously obtained candidate segmentation points. This strategy has some similarities with Kashino's beat and terminal point integration [Kashino *et al.,* 1995]. The main difference is that, there, beats are used in the definition of global processing scopes, rather than note onsets. However, if only the main beats are used, consecutive notes at the same pitch coming out during the interval between two beats may pass undivided. Also, beats occurring in the middle of a note may induce inadvertent segmentations.

A. Onset Matching

Hence, onset matching is carried out for all candidate segmentation points. Namely, if a candidate valley is close to an identified onset, i.e., they are separated by less than *maxValleyOnsetDiff* = 20 msec, that valley is defined as an actual segmentation point.

B. Note Segmentation and Time Correction

Next, the collected segmentation points are used to further split the notes that resulted from the frequency-based segmentation stage. Here, the determined points are adjusted to the locations of the detected onsets.

Moreover, the starting times of the notes are also adjusted when onsets are found close to the original note beginnings (with a maximum *maxValleyOnsetDiff* distance, and occurring before the start of the note). This copes to some extent with the problem of noisy attack transients, where the pitch may oscillate significantly. In such cases, the track receives no F0s from the note attack region, causing late note beginnings.

Onsets located after the original note beginnings are not taken into consideration since this would give rise to note shortening, which, if inadvertently performed, is not recoverable later on. On the contrary, too long notes may be corrected in the melody

extraction stage, when note truncation is applied, as will be seen in Section 5.3.2.

C. Illustration and Remarks on the Validation Approach

The results of salience-based segmentation for an excerpt from Claudio Roditi's "Rua Dona Margarida" are illustrated in Figure 4.12. Gray horizontal lines represent annotated notes, whereas black lines stand for the extracted notes. The small gray vertical lines denote the desired segmentations and the black vertical ones represent the obtained segmentation points.

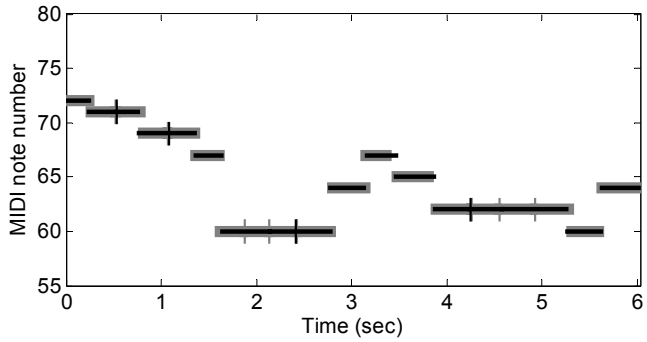


Figure 4.12. Results of the salience-based track segmentation algorithm.

It can be seen that the detected segmentation points match almost perfectly the annotated ones. However, four true segmentations are missing. Ideally, our method would catch such absent points if perfect onset detection were achieved. As this is not the case, "clear" segmentation points without matching onsets are erroneously left out.

In order to improve the results, we chose to redo salience-based segmentation after melody identification (described in Chapter 5), with a different set of restrictions. Thus, "clear" segmentation points (quantitatively defined in the next subsection) are selected no matter if no corresponding onsets are found. The reason why we did not proceed in this way at this stage was motivated by the fact that excessive segmentation decreases the global melody detection accuracy (due to the decision-making mechanisms involved in the resolution of note overlapping, in Section 5.3). On the other hand, postponing unclear segmentation cases to the end of the chain improves segmentation results, while keeping overall melody accuracy, which was experimentally confirmed.

4.4.4. Segmentation after Melody Identification

We repeated salience-based segmentation after melody identification, where we accepted

all clear minima in the salience sequence as valid segmentation points, i.e., valleys whose prominences are above *clearValleyProm* = 35 (recall the normalization to the [0; 100] interval, in the pitch detection stage). The attained results are presented in Figure 4.13.

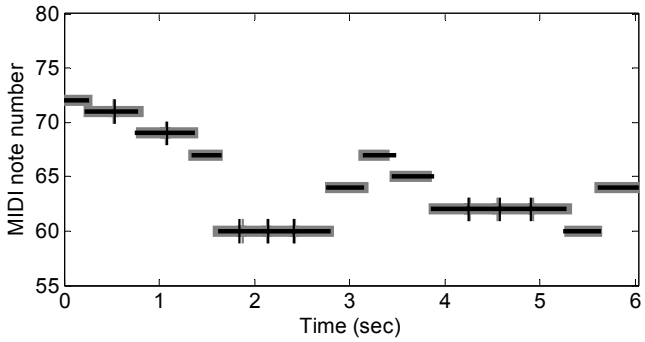


Figure 4.13. Results of the salience-based track segmentation algorithm: acceptance of clear salience minima.

It can be seen that there is an almost perfect match when this solution is followed. However, in some excerpts excessive segmentation occurs, especially when significant amplitude modulation is present. A solution to this problem would require more robust onset detectors in polyphonic contexts. Anyway, by resynthesizing the extracted melody, we got the subjective perception that melodies were easier to recognize in over-segmented cases than in under-segmented situations.

4.4.5. Putting It All Together

The operations carried out for salience-based segmentation are summarized in Algorithm 4.3. Parameter definition is presented in Table 4.3.

Algorithm 4.3. Salience-based track segmentation.

```
1. Detect candidate segmentation points:
   1.1. Apply ZOH to the original salience sequence.
   1.2. Look for clear salience minima, complying with the de-
        scribed recursive procedure.
2. Perform onset detection directly on the audio signal:
   2.1. Implement band-Pass filtering.
```

```
    2.2. Compute the energy variations in each band, based on the
         half-wave-rectified first derivative of the amplitude
         envelope in each band.
    2.3. Integrate the information in all bands and detect onsets
         (linear summation, peak detection and deletion of onset
         candidates in dense regions).
3. Validate candidate segmentation points:
    3.1. Match detected onsets to the original segmentation can-
         didates.
    3.2. Adjust the timing of the segmentation points, as well as
         the start of the notes, to the found onsets.
4. Repeat salience segmentation after melody identification:
    4.1. Accept all clear minima in the salience sequence (i.e.,
         whose prominence is above 35 units).
5. Return the resulting segmented notes.
```

Parameter Name	*Parameter Value*
minPvd	10% of amplitude range
filterbank frequency range	[44, Nyquist frequency] Hz
filter types	elliptic
filter order	3rd (6th in practice)
band-pass ripple	1.5 dB (≈3 dB in practice)
stop-band attenuation	20 dB (≈40 dB in practice)
decimation frequency	200 Hz
half-Hanning duration	100 msec
minPeakSal	0.05
minOnsetDiff	50 msec
clearOnsetMag	0.4
maxValleyOnsetDiff	20 msec
clearValleyProm	35

Table 4.3. Parameters for salience-based track segmentation (line 1: detection of candidate segmentation points; lines 2 to 11: onset detection; line 13: validation of candidate points; line 14: segmentation after melody identification.

4.5. Putting It All Together

The entire methodology for conversion of pitch sequences into musical notes is summed up in Algorithm 4.4.

Algorithm 4.4. From pitches to notes.

1. Perform pitch trajectory construction, as in Algorithm 4.1:
   1.1. Quantize frequencies to MIDI note numbers.
   1.2. Implement peak continuation based on frequency proximity, allowing track inactivity and tackling ambiguities through the developed look-ahead procedure.
   1.3. Eliminate short tracks.
   1.4. Reassign unused pitch candidates to validated tracks.
2. Perform frequency-based segmentation, as in Algorithm 4.2:
   2.1. Conduct note segmentation, namely regarding oscillation filtering, filtering of delimited sequences and glissando filtering.
   2.2. Adjust the precise timings of each note candidate.
   2.3. Assign a MIDI note number to each note candidate, coping with off-key performances or tuning variations.
3. Execute salience-based segmentation, as in Algorithm 4.3:
   3.1. Define a set of candidate segmentation points, corresponding to minima in the pitch salience sequence.
   3.2. Perform onset detection directly on the raw audio signal.
   3.3. Validate the previous candidate segmentation points by matching them with the obtained note onsets.
   3.4. Segment the note candidates with the defined segmentation points.
   3.5. Repeat step 3.1 after the melody identification stage (see Chapter 5), selecting all clear salience minima as valid segmentation points.
4. Return the resulting segmented notes.

4.6. Experimental Results, Analysis and Conclusions

The results of note detection are presented and discussed in the next paragraphs, namely in terms of trajectory creation and frequency and salience-based track segmentation. The main encountered difficulties are addressed and possible improvements are suggested.

A. Analysis of Results

Summary results are presented in Table 4.4. In the first column (after the ID column), the results of pitch detection are repeated for ease of comparison. The next columns show raw pitch accuracy after PTC and track segmentation.

By comparing the first two columns, it can be seen that after the creation of pitch tracks, most of the originally detected pitch candidates are still kept[52]. An average loss of 1.1% has however occurred, most notably in the male opera expert (ID 19), where 11.8% of the initially found pitch candidates were lost. In effect, the SACF was particularly noisy in this track, with several close peaks. This complicated the track building process in such a way that the look-ahead scheme could not disentangle. Even so, pitch trajectories were generally well constructed.

As regards note determination, the overall melodic note accuracy raised by almost 10%, as can be seen in the last column. This is a result of quantizing the extracted and annotated F0s to the corresponding ETFs, plus filling in inactive frames. In fact, even originally small pitch errors (of, say, 5 cents) disappear after quantization. Moreover, in our test-bed the previously mentioned glissando difficulties are resolved since all extracted F0s are now assigned a value corresponding to the ETF of the identified note. This is particularly notorious in Ricky Martin and Avril Lavigne's excerpts (IDs 5 and 6), which improved by around 20%.

Furthermore, missing values in the initial pitch tracks (as a consequence of track sleeping) are now filled in with the quantized F0 values. The problem of missing values during the attack of the notes is also tackled by adjusting note beginnings with recourse to the detected onsets. Excerpt 17 (midi2) is an example of a generous improvement that resulted from this procedure, added to the correction of pitch deviations.

Nevertheless, one problem occurred in the pop1 (20) excerpt, where the accuracy decreased just about 6%. This was due to a few semitone errors that came from the note labeling stage. In any case, by comparing the values of this sample in the last two columns, we can notice that tuning compensation was not responsible for these errors, as will be discussed in the next paragraph. The same semitone difficulty occurred in the opera excerpts. However, in these, the accuracy improvements from quantization (a few

[52] As in the pitch detection stage, quantized frequencies were used in our database, whereas in the M04 set exact extracted and annotated frequencies were employed.

frames had F0 errors of approximately 40 cents) and the resolution of missing values counterbalanced the generated semitone errors, particularly in the male excerpt (19).

ID	*AMPD (MRPA)*	*PTC (MRPA)*	*TS (no compens.) (MRNA)*	*TS (MRNA)*
1	97.0	96.8	98.0	98.0
2	75.8	75.7	82.2	82.2
3	89.4	89.1	95.4	95.4
4	76.4	76.2	96.6	96.6
5	62.3	61.5	84.9	85.3
6	75.2	75.2	93.6	93.6
7	95.7	95.7	98.8	98.8
8	91.7	91.0	93.6	93.6
9	55.0	54.7	63.9	86.5
10	70.9	69.9	81.1	81.1
11	92.4	92.4	95.4	95.4
12	89.7	89.4	96.5	96.5
13	93.6	92.4	98.1	98.1
14	76.5	75.4	82.1	81.1
15	81.7	81.2	90.8	90.8
16	80.8	80.2	92.2	91.7
17	73.7	73.5	98.4	98.4
18	79.7	77.6	76.0	79.6
19	75.1	63.8	69.5	70.2
20	80.3	78.3	72.5	72.7
21	88.2	87.1	86.9	87.0
Avg PDB	*80.2%*	*79.8%*	*89.4%*	*91.5%*
Avg M04	*81.9%*	*79.9%*	*86.3%*	*86.6%*
Avg	*81.0%*	*79.9%*	*87.9%*	*89.2%*

Table 4.4. Note determination results: accuracy for PTC and trajectory segmentation, with and without tuning compensation.

Regarding note labeling, and because of tuning compensation, the overall average

results increased from 87.9% to 89.2% (third and fourth columns, respectively). As can be seen in the last column of Table 4.4, most excerpts were not affected by the adjustment scheme, in a sign that the median frequency values did not deviate significantly from the ETFs. In only two samples (IDs 14 and 16), the labeling step had a slight negative effect (a decrease of 1% or less), whereas in other samples, the results improved a little (e.g., IDs 5, 18, 19, 20 and 21 – from 0.2 to 3.6%). Nonetheless, the applied method was responsible for a substantial improvement in the results of the Eliades Ochoa's excerpt (ID 9), which increased from 63.9 to 86.5%.

ID	*# Tracks to Segm.*	*% False Negatives*	*Time Error*	*# Extracted*	*% False Positives*	*% Semitone Errors*
1	3	0.0	9.2	16	0	0
2	1	0.0	16.5	13	0	0
3	0	(--)	(--)	11	0	0
4	2	0.0	85.8	16	0	0
5	1	0.0	15.1	10	0	10
6	3	0.0	10.6	14	0	0
7	2	0.0	33.5	19	0	0
8	3	0.0	32.6	12	0	0
9	1	0.0	24.8	10	0	0
10	0	(--)	(--)	11	0	0
11	4	25.0	31.3	26	0	0
12	2	0.0	18.5	23	0	0
13	0	(--)	0.0	11	0	0
14	2	0.0	94.2	21	0	4.8
15	0	(--)	(--)	22	0	0
16	3	0.0	19.4	38	0	0
17	0	(--)	(--)	22	0	0
18	8	37.5	17.3	37	0	10.8
19	10	80.0	12.2	52	0	9.6
20	4	50.0	5.6	34	5.88	11.8
21	1	0.0	44.1	28	0	0
Sum / Avg PDB	*20*	*5%*	*28.7 msec*	*158*	*0.0%*	*0.6%*
Sum / Avg M04	*30*	*43.3%*	*27.2 msec*	*288*	*0.7%*	*4.9%*
Sum / Avg	*50*	***28.0%***	***28.0 msec***	*446*	***0.45%***	***3.4%***

Table 4.5. Results for frequency-based track segmentation.

Detailed results for frequency-based segmentation are presented in Table 4.5. The

first column represents the number of pitch tracks that are necessary to segment after PTC. The second column denotes the percentage of missed segmentations and the third one presents the average time error for the obtained segmentation points in comparison to the annotated ones. The fourth column shows the total number of extracted notes for each song excerpt, from which the percentage of false positives and semitone errors are computed (last two columns). Results are summarized in the bottom lines, where the values in "#" columns are summed, whereas all the others are averaged.

In the comparison of extracted and annotated segmentation points, we defined a maximum time deviation of *max* {*maxOnsetDist*[53], 10% of pitch track length}. This is the reason why timing errors in the order of 90 msec were present in a few samples.

The timing accuracy was generally good (an average error of 28 msec), with most deviations under 20 msec. These may have even resulted from annotation inaccuracies. A few slightly higher deviations occurred in tracks with transition zones with many empty frames. However, two particularly high time errors appeared in one track from Dido (ID 4) and another one from the jazz2 excerpt (ID 14). There, an interesting situation occurred: the pitch track contained a solo note, which was followed by an accompaniment note and again a note solo. The accompaniment note was the one responsible for the inclusion of two notes in the same pitch track. Then, the most notorious minima in the composite salience sequence did not correspond to the actual note boundaries. This situation gives a good illustration of some of the difficulties entailed in the accurate construction of pitch tracks in polyphonic contexts.

As for the detection of segmentation points, very good results were achieved in the PDB database (5% false negatives). Nevertheless, the average performance in the M04 database was poor, where 43.3% of the required segmentations passed undetected. This was mostly due to the opera excerpts. In reality, the frequency-based segmentation strategy had some troubles with samples with extreme vibrato. In these, the first stage of MIDI quantization led to a succession of several short initial PCFs. Since the method relies on the detection of long PCFs, the resulting ambiguities were not properly disentangled. Moreover, many empty frames existed in those tracks, placing additional difficulties on the algorithm. This was particularly problematic in the male excerpt, where the SACF was particularly noisy. However, if the opera samples are excluded from analysis, an average of 16.7% turns out, which is more acceptable. Indeed, in all the other excerpts most of glissando and frequency-modulated notes were correctly dealt with.

In terms of false positives, only two notes were inadvertently segmented, namely in the pop1 (20) excerpt. In effect, the implemented approach was somewhat more prone to false negatives than to false positives.

Regarding semitone errors, most of them occurred in the opera excerpts. We are es-

[53] *maxOnsetDist* = 62.5 msec, as defined and motivated in the next chapter (see Table 5.1).

pecially satisfied with the fact that in 446 extracted notes involving complex dynamics and tuning issues, only 15 semitone errors (3.4%) occurred.

To sum up the analysis on frequency segmentation accuracy, we have also calculated average precision and recall figures. The average recall (i.e., the percentage of annotated segmentation points correctly identified) was 72% and the average precision (i.e., the percentage of identified segmentation points that corresponded to actual segmentation points) was 94.7%.

ID	*# Tracks to Segm.*	*% False Negatives.*	*Time Error*	*# Extracted*	*% False Positives*
1	0	(--)	(--)	16	0.0
2	7	57.1	29.0	13	7.7
3	5	60.0	46.1	11	18.2
4	2	0.0	25.8	16	6.3
5	1	0.0	11.1	10	60.0
6	3	0.0	26.3	14	7.1
7	8	0.0	17.0	19	0.0
8	1	0.0	3.2	12	8.3
9	3	0.0	20.9	10	0.0
10	4	50.0	9.2	11	9.1
11	5	40.0	27.6	26	0.0
12	1	0.0	56.4	23	13.0
13	0	(--)	(--)	11	27.3
14	3	33.3	12.8	21	9.5
15	3	0.0	3.1	22	9.1
16	2	0.0	41.7	38	5.3
17	6	0.0	35.7	22	45.5
18	6	33.3	52.8	37	29.7
19	6	50.0	48.3	52	38.5
20	1	0.0	10.1	34	14.7
21	5	20.0	43.3	28	21.4
Sum / Avg PDB	*39*	*28.2%*	*22.3 msec*	*158*	*8.2%*
Sum / Avg M04	*33*	*21.2%*	*35.7 msec*	*288*	*22.2%*
Sum / Avg	*72*	***25.0%***	***28.8 msec***	*446*	***17.3%***

Table 4.6. Results for salience-based track segmentation.

Most of the encountered difficulties came from opera tracks with extreme vibrato. In

those cases, the number of false negatives and semitone errors was clearly above the average. In any case, in excerpts with moderate vibrato, results were quite satisfactory.

Regarding salience-based segmentation, results are presented in Table 4.6.

As expected, this is quite complex in polyphonic contexts, since onset detection is not trivial in this case. Furthermore, harmonic collisions between different sources corrupt the energy levels of F0 candidates, in which the algorithm bases the identification of segmentation candidates. Also, the distinction between amplitude modulation and note boundaries is not always clear. In addition, reverberation effects such as the ones found in Enya's excerpt add extra difficulties to the problem.

Nevertheless, 25% of false negatives and 17.3% of false positives can be considered acceptable for this stage of research. Further complexities were found in Hallelujah (ID 2), Enya (3), Juan Luis Guerra (10), Battlefield Band (11) and male opera (19), where the number of false negatives was clearly above the average. These generally correspond to samples with higher polyphonic complexity, reverberation or tremolo (opera). In other complicated excerpts, such as Eliades Ochoa's, the algorithm was rather successful.

In some other excerpts, false negatives did not occur significantly but false positives appeared instead, e.g., in Ricky Martin (5) and midi2 (17). In reality, the balance between under and over-segmentation proved difficult to achieve.

With respect to timings, some deviations of around 40-50 msec occurred, which are higher than desired, though not excessively. This happened partly because the exact locations of valleys in the pitch salience sequences are disturbed by harmonic collisions with sonic components from other sources. Slight annotation errors also affected it.

To sum up the salience-based segmentation phase, as the number of identified segmentation points was high in comparison to the actual segmentation points, the average precision was low: 41.2%. With respect to recall, an average of 75.0% was accomplished.

In order to evaluate the influence of parameter values in the variance of the results, the employed thresholds were individually modified, typically in a [-50%, +50%] range from the assigned values. Here, we did not evaluate the results in terms of precision and recall (due to the partly manual evaluation that was necessary to conduct at this stage, e.g., in the inspection of the created trajectories, so as to identify the required segmentation points). Instead, the overall outcome on melody identification was measured. Here, a maximum average decrease of 6% was observed in the MRNA metric. However, a few individual excerpts had higher variations. For instance, in Juan Luis Guerra's sample (ID 10) we noticed performance oscillations of up to +5% and –15%.

Of particular interest is the *minNoteLen* parameter, as this is the main factor involved in frequency-based segmentation. A range of values between 60 and 150 msec was experimented, where the maximum perceived decrease in the average melody note accuracy was 6.5%, stemming from a minimum note duration of 60 msec. Moreover, best results were obtained with the defined threshold (125 msec). It is noteworthy that this value is

close to the 150 msec referred by Bregman [Bregman, 1990, pp. 462]. Although we did not compute precision and recall figures, we confirmed that this value also led to the most accurate frequency-based segmentation (by visual inspection).

Finally, in the MIREX'2004 evaluation, except for our algorithm only Bello's carried out note segmentation. An edit distance score was calculated, in which our method behaved generally better as a direct consequence of its overall higher pitch detection accuracy [Gómez *et al.,* 2006; MIREX, 2004]. Nonetheless, even in several cases where Bello's algorithm had a better pitch detection performance, our approach still yielded a better edit distance score (see Section 2.6.2). This is a result of Bello's mechanism for note determination. Although not many details on the pursued strategy are provided [Gómez *et al.,* 2006], we observed that over-segmentation frequently occurred in his method, which probably resulted from too low maximum trajectory sleeping. Hence, a profusion of fragments corresponding to the same note arises. These repetitions are punished as wrong insertions by the edit distance metric, increasing the computed distance.

B. Limitations of the Algorithm and Possible Improvements

A difficulty in pitch trajectory construction derives from the fact that accompaniment notes may be included in melodic pitch tracks because of frequency continuation. This is a complex problem, since timbre is hard to "measure" and, thus, sources are not recognized before peak continuation. In any case, when this situation causes sudden intensity differences, salience-based segmentation can handle it to some extent.

Likewise, melodic notes may be continued by harmonics of other notes. This situation further motivates the need to solve the issue of harmonically-related peaks during pitch detection. Again, salience-based segmentation attenuates this difficulty.

Also, the question of closely spaced F0 candidates gave rise to some difficulties in the PTC process, namely in the male opera excerpt. Once more, this could be tackled complying with the lines suggested in the pitch detection stage (Section 3.6).

Regarding track inactivity, when F0s are missing the overall energy level in the note's frequency range should be analyzed. For the sake of accuracy, a track should only be allowed to sleep if the energy level in its frequency range was sufficient to presume masking. Otherwise, no perceptual restoration would have occurred, being more likely that the note had actually stopped.

As for frequency-based segmentation, the main difficulties of the followed methodology resulted from its dependency on the *minNoteLen* parameter. Indeed, pitch tracks with extreme vibrato were sometimes hard to accurately segment. This is a more fundamental issue, which would probably require a different PCF identification approach. Namely, instead of quantizing the F0 values to MIDI note numbers, the original frequency curve should be directly analyzed. However, algorithms for piecewise function approximation seemed inadequate to our problem, besides being rather parame-

ter-dependent.

Finally, the results achieved for salience-based segmentation are encouraging but the balance between over and under-segmentation needs additional attention. Namely, more robust polyphonic onset detectors are required. Also, the onsets of accompaniment notes may mislead the procedure for validation of segmentation candidates. A possible way of improving the method would be to avoid validations by onset candidates that match the beginnings of other detected notes. Moreover, the pitch salience sequence is corrupted by harmonic collisions, especially in excerpts with higher polyphonic complexity. This could be attenuated by an iterative estimation and cancellation scheme for pitch detection, as referred to in Section 3.6.

As regards the discrimination between note boundaries and amplitude modulation, this should be further worked out. In fact, Albert Bregman suggests that abrupt rises in intensity represent new notes [Bregman, 1990, pp. 71]. Therefore, if amplitude modulation is slow, the succession of pitches is simply heard as a single note. However, when amplitude modulation is fast, the higher rates of intensity growth promote the perception of several consecutive notes. In this way, the slope of salience variation until steady-state is reached should be measured and used to validate segmentation candidates.

Chapter 5

IDENTIFICATION OF MELODIC NOTES

"Figure-ground relationships constitute the main dialectic of our European-North American music culture and can be found in the dualism between melody and accompaniment."

Philip Tagg, "Reading Sounds", 1986

In the final stage of the present melody detection system, our goal is to identify the notes that convey the main melodic line. As a result of the previous stages, several notes are created, among which the melody must be isolated.

The separation of the melodic notes in a musical ensemble is not a trivial task. In effect, many features of auditory organization influence the perception of the main melody by humans, for instance in terms of the pitch, timbre and intensity content of the instrumental lines in the sonic mixture. Moreover, factors such as selective attention, experience or personal interest, are involved in figure-ground organization as well.

In the algorithm described in this chapter, we have made particular use of intensity and frequency proximity aspects. As the conducted approach induces the selection of accompaniment notes when the melody is absent, we also aim to remove such notes.

Section 5.1. Introduction

We begin this chapter with a review of the existing methodologies for identification of melodic notes (or melodic pitch lines) in an ensemble.

Section 5.2. Elimination of Ghost Harmonically-Related Notes

Our algorithm starts with the elimination of ghost harmonically-related notes, which are a consequence of selecting both true pitches and sub or super-harmonics during pitch detection. In order to dispose of such notes, we make use of the perceptual rules of sound organization designated as harmonicity and common fate.

Section 5.3. Selection of the Most Salient Notes

As an initial attempt towards melody identification, we select the most salient notes among the ones present after ghost note elimination. Here, non-dominant notes (where dominance is defined according to note intensity level) are deleted, as well as low-frequency notes.

Since the remaining notes are not allowed to overlap in time, such note overlaps are resolved. Basically, a note might either be removed or truncated.

Section 5.4. Melody Smoothing

The previous scheme has some limitations, as the notes comprising the melody are not always the most salient ones. This is particularly clear when abrupt pitch transitions occur. Thus, we improved the method by smoothing out the melody contour.

To this end, octave-correction is first employed, since not all ghost notes were previously eliminated.

Next, we handle abrupt pitch jumps by defining regions of smoothness and correcting situations where the extracted melody moves suddenly to different pitch registers.

Finally, since erroneous notes are discarded, we fill in the gaps with notes that are more likely to belong to the main melody. As a result of removing erroneous notes, the original timings of previously truncated notes are restored as much as possible.

Section 5.5. Elimination of Spurious Accompaniment Notes

In the previous steps, the algorithm outputs the most salient notes at each time that guarantee a smooth melody contour. Consequently, notes from the accompaniment may turn up. Particularly, spurious accompaniment notes may appear when pauses between melody notes are sufficiently long. Hence, we dispose of such notes by looking for sudden intensity or duration variations in the sequence of melodic notes.

Section 5.6. Note Clustering

Besides the previous case, notes from the predominant accompaniment are also output when the solo stops. In reality, it is common that a secondary instrument takes the lead during the time intervals when the melody is silent. In this way, we aim to discriminate between true melodic notes and false positives through note clustering. Here, a number of acoustic features are extracted and fed in to a Gaussian Mixture Model that attempts to separate the melody from the accompaniment. In addition, dimensionality reduction is performed, recurring to Principal Component Analysis and forward feature selection.

Section 5.7. Putting It All Together

The entire melody identification algorithm is summarized and the defined model parameters are listed in this section.

Section 5.8. Experimental Results, Analysis and Conclusions

Finally, experimental results are presented and analyzed. The main benefits and shortcomings of the proposed mechanism are discussed and pointers for future improvements are provided.

5.1. Introduction

The identification of the notes bearing the melody of a song, being probably the central task of any melody detection algorithm, is also one of the most difficult to carry out.

The perception of melody is related to the phenomenon of figure-ground organization, as referred to in Section 2.2.2. Furthermore, according to our context, the main melody in a song may emerge from the identification of the most prominent musical part, which clearly stands out from a more diffuse, less organized or less interesting background. Thus, melody extraction can be regarded as a problem of source separation.

5.1.1. Approaches based on Full Source Separation

Full sound source separation is an important issue for polyphonic music analysis and automatic music transcription. However, computational sound-source recognition and, particularly, separation has proved to be very hard. Some attempts have been conducted in that direction with no general neither accurate results so far. Namely:

i) techniques inspired on computational auditory system analysis [Ellis, 1996];

ii) rule-based methodologies, targeting especially musical signals, which take advantage of voice-leading rules of music composition [Temperley, 2001];

iii) local optimization approaches, such as the one in [Kilian and Hoos, 2002] for voice separation in music in symbolic notation, where a cost function, making particular use of pitch and gap distances, is optimized;

iv) and data-adaptive techniques, a.k.a. blind source separation, where there is no knowledge of the sources present, which induces to their estimation based solely on the available data with recourse to techniques such as Independent Component Analysis [Casey and Westner, 2000; Smaragdis, 2001] or assuming sparseness of the sources [Plumbey *et al.*, 2001; Virtanen, 2003].

The first three approaches are intimately related, since the usually employed voice-leading rules and cost metrics strongly rely on perceptual rules of sound organization [Huron, 2001]. However, without exploiting timbre information (which is hard to acquire unequivocally, as will be discussed), several difficulties arise, for example when performing separation based on rules such as "avoid crossing of voices". Therefore, data-adaptive techniques seem more appealing here, since they aim to unscramble the mixture into its constituent time-domain signals without any explicit assumptions on sound organization or model parameters. This, in turn, would allow for subsequent monophonic pitch detection. Nevertheless, the accuracy of data-driven techniques in polyphonic real-world signals is much too low.

5.1.2. Approaches based on Figure-Ground Organization

On the other hand, as far as melody identification is concerned, full source separation is not the ultimate goal. Instead, the objective is to separate the melody from "all the rest". In this way, the problem of complete source separation is usually not tackled. Instead, the notes (or the pitch line) conveying the melody are separated from the background accompaniment, in compliance with the lines of figure-ground organization.

This figure-ground approach is inherent to most of the melody detection systems that extract several pitch candidates in each frame. As discussed in Section 2.4.2, Goto defines the main melodic line as the one corresponding to the most predominant agent, utilizing specific salience and reliability measures [Goto, 2000]. In [Marolt, 2005], melodic seeds are first obtained recurring to the loudest fragments, after which clusters are formed based on similarity metrics. In the melodic-path-founding strategy devised in [Eggink and Brown, 2004], F0 strength, i.e., the intensity of each F0 candidate, proved to be the most important knowledge source, although the instrument recognition module played an important role on the selection of the notes representing to the solo. In Bello's method [Gómez *et al.*, 2006], the melodic and non-melodic fragments that result from peak continuation are discriminated using a rule-based system: the melodic path is the one that maximizes the energy while minimizing steep changes in the tonal sequence. Dressler also defines the melodic pitch line following a rule-based scheme, where tone successions containing intervals larger than the octave are avoided and notes from middle or higher pitch registers are preferred [Dressler, 2005]. In [Ryynänen and Klapuri, 2005b], the optimal melodic path is found with recourse to a musicological model where between-note transition probabilities are employed.

We founded our algorithm on the assumptions that i) regarding intensity, the main melodic line often stands out in the mixture (salience principle) and that ii) melodies are usually smooth in terms of pitch intervals (melodic smoothness principle). Moreover, although it can be argued that in the perception of melody the dominant accompani-

ment is perceived as figure when the solo is absent, we defined melody in a stricter sense, i.e., only the sequence corresponding to a solo instrument, as referred to in Section 2.3. Hence, we conducted some efforts to dispose of false positives, i.e., non-melodic notes in the resulting melody, by spurious note removal and clustering.

5.2. Elimination of Ghost Harmonically-Related Notes

The set of candidate notes resulting from trajectory segmentation typically contains several *ghost harmonically-related notes*. The frequency partials in each ghost note are actually multiples of the true note's F0, if the ghost note is higher than the true note, or submultiples, if it is lower. Therefore, the objective of this step is to get rid of such notes.

In short, we make use of the perceptual rules of sound organization designated as harmonicity and common fate, described in Section 2.2.2 [Bregman, 1990, pp. 227-292]. Namely, we look for pairs of harmonically-related notes with common onsets or endings and with common modulation, i.e., whose frequency and salience sequences change in parallel. We then delete the least salient note if the ratio of its salience to the salience of the other note is below a defined threshold.

5.2.1. Exploiting Harmonicity

In the harmonicity rule, if two note candidates occurring in a common time window have ETFs such that one is a multiple of the other[54], it is possible that both frequency sequences denote harmonics of the same note. In this case, one of the notes might have resulted from the selection of super/sub-harmonics of the F0 in the pitch detection stage. In order to deal with possible semitone errors, a tolerance of ± 1 semitone is admitted in the comparison of such note candidates.

As to the mentioned "common time window", some relaxation is introduced by allowing a maximum separation between the beginnings and endings of the note candidates under comparison. In quantitative terms, two notes are said to have common onsets if their beginnings differ at most by *maxOnsetDist* msec, which was set to the same value as the *maxSleepLen* parameter, i.e., 62.5 msec. The same maximum difference applies when comparing the two notes' endpoints. This somewhat high value was experimentally set, so as to handle timing inaccuracies that may result from noise and frequency drifting at the beginnings and endings of notes. In actual notes, the onset asyn-

[54] In case we only compared notes separated by an octave, some ghosts would pass undetected. Illustrating, for a note whose ETF is 220 Hz, we wish to check all its multiples, e.g., 220, 440, 660 and 880 Hz. If only octaves were considered, a hypothetical ghost note at 660 Hz would not be evaluated.

chronies between partials do not exceed 30 to 40 msec, since at that point each partial may be heard as a separate tone [Handel, 1989, pp. 214].

The detected pairs of harmonically-related note candidates are then analyzed, as there is a possibility that one of them is a ghost note. In order to check this hypothesis out, we evaluate the so-called common fate rule, as well as the salience of the two notes.

5.2.2. Exploiting Common Fate

In the common fate rule, harmonically-related frequency sequences can be grouped by taking advantage of aspects such as common modulation, both in frequency and in amplitude. Indeed, components belonging to the same note tend to have synchronized and parallel changes in frequency and intensity (here represented by pitch salience). Hence, we measure the distance between frequency curves for pairs of harmonically-related note candidates. Similarly, we calculate the distance between their salience sequences.

Formally, the distance between frequency curves is calculated as in (5.1), based on [Virtanen and Klapuri, 2000]:

$$d_f(i,j) = \frac{1}{t_2 - t_1 + 1} \sum_{t=t_1}^{t_2} \left(\frac{f_i(t)}{avg(f_i(t))} - \frac{f_j(t)}{avg(f_j(t))} \right)^2 \tag{5.1}$$

where d_f, represents the distance between two frequency trajectories, $f_i(t)$ and $f_j(t)$, during the time interval $[t_1, t_2]$ where they both exist. The idea of (5.1) is to scale the amplitude of each curve by its average, thus, normalizing it. Simple mean subtraction is insufficient here, as the frequency curves of different harmonics have different ranges. The expression in (5.1) is similar to normalized correlation, which could have been alternatively employed. An identical process is followed for the salience sequences.

This procedure is illustrated in Figure 5.1 for the frequency sequences of two harmonically-related note candidates from an opera excerpt with extreme vibrato (opera female 2 in Table 2.1). We can see that the normalized frequency curves are very similar, which provide good evidence that the two sequences are both part of the same note.

Additionally, we found it beneficial to measure the distance between the normalized derivatives of frequency curves as well (and, likewise, the derivatives of salience sequences). In fact, it is common that these curves have high absolute distances showing, however, the same trends. The distance between derivatives is used as another measure of curve similarity. This is illustrated in Figure 5.2 for the pitch salience sequences of two notes from the same opera excerpt. It can be seen that, although the depicted saliences differ somewhat, their trends are very similar, i.e., the distance between the normalized derivates is small. In this way, it is also likely that they both belong to the same note.

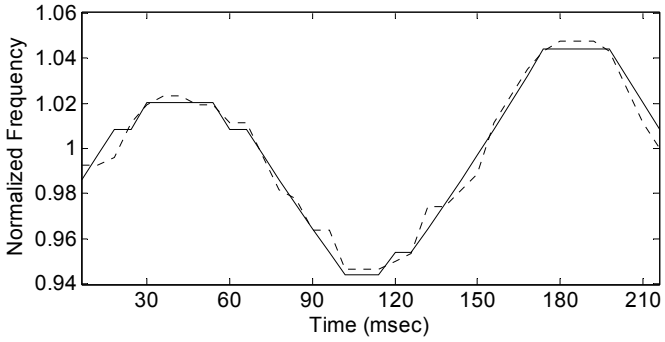


Figure 5.1. Similarity analysis of frequency curves.

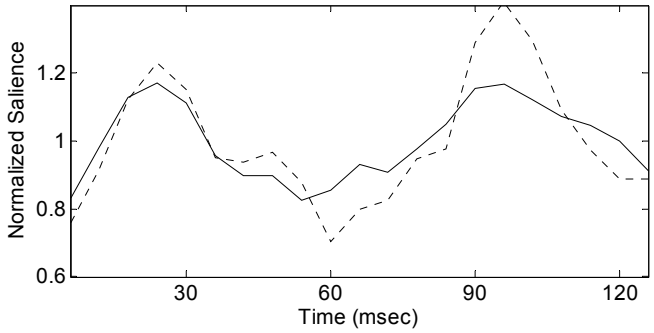


Figure 5.2. Similarity analysis of salience trends.

To conclude the analysis on common modulation, we assume that the two frequency sequences have parallel changes if any of the four computed distances (in frequency, salience, or their derivatives) is below a threshold of 0.04, defined in the *maxAllowedCurveDist* parameter.

5.2.3. Integration of Harmonicity and Common Fate

Finally, we compare the saliences of pairs of harmonically-related notes that satisfy the common fate requirement in order to take a decision: if the salience of one of the notes is much lower than the other's, the least salient one is eliminated. Quantifying, a note is removed if its salience is less than 40% the one of the most salient note in case they are separated by an octave (set in the *minBasicSalienceRatio* parameter), 20% in case the ETF of the highest note is the triple of that of the lowest one, and so forth.

5.2.4. Putting It All Together

This algorithm is summarized in Algorithm 5.1. Parameter definition is presented in Table 5.1.

Algorithm 5.1. Elimination of harmonically-related notes.

1. Sort all notes in ascending onset time order.
2. For each note, *i*:
   2.1. Look for a note, *j*, such that:
      a) (|onset(*i*) – onset(*j*)| ≤ *maxOnsetDist* or |ending(*i*) – ending(*j*)| ≤ *maxOnsetDist*) and
      b) ETF(MIDI(*i*, *i* ± 1)) is a (sub-)multiple of ETF(MIDI(*j*)) (the ratio, *r*, of the highest ETF to the lowest is calculated) and
      c) the two notes have parallel changes in frequency and salience.
   2.2. If note *j* was found,
      2.2.1. Compute the average salience of the two notes in their common time interval, avgSal.
      2.2.2. If avgSal(*j*)/avgSal(*i*) ≤ *minBasicSalienceRatio*/(*r*-1) (i.e., 0.4/(*r*-1)) then
         - delete note *j* and repeat step 2.1 until no more notes are found.
      2.2.3. If avgSal(*i*)/avgSal(*j*) ≤ *minBasicSalienceRatio*/(*r*-1) then
         - delete *i* and repeat step 2 for the next note.
3. Return the reduced set of notes.

Parameter Name	*Parameter Value*
maxOnsetDist	*maxSleepLen* (62.5 msec)
maxAllowedCurveDist	0.04
minBasicSalienceRatio	0.4

Table 5.1. Parameters for the elimination of ghost harmonically-related notes.

The values for curve distance and salience ratio thresholds were experimentally set so that the elimination of true melodic notes was minimal, while still deleting a substantial amount of ghost notes. This is motivated by the fact that missing notes cannot be recovered in later stages but false candidates can be eliminated afterwards.

In the used database, an average of 37.8% of the notes resulting from the note determination stage were removed. Moreover, only 0.3% of true melodic notes were inadvertently deleted. Although many ghost notes are discarded at this point, a high number of non-melodic notes is still present. Namely, only 25.0% of all notes belong to the melody. This poses interesting challenges to the next steps of the algorithm.

Exemplifying, in the same excerpt from "Rua Dona Margarida" used before, 57.9% of the notes are eliminated (55 out of 95). From the remaining 40, only 19 should be selected as making part of the final melody. This point is illustrated in Figure 5.3.

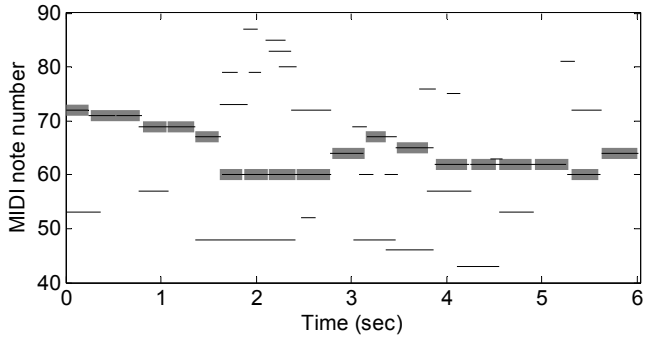


Figure 5.3. Results of the elimination of ghost notes.

As before, the black lines denote the resulting notes after elimination (i.e., the 40 remaining notes), whereas the gray horizontal lines represent the original annotated notes (19 notes; some of the lines correspond to sequences of several notes at the same pitch, whose intervals are not noticeable due to the graphic resolution). As can be seen, in this example all the melodic notes are present and the task of the algorithm is now to identify them among the entire set of notes.

5.3. Selection of the Most Salient Notes

As previously mentioned, intensity is an important cue in melody identification. Therefore, we select the most salient notes as an initial attempt to melody detection. We have also evaluated the possibility of choosing the notes with highest frequency, as suggested in [Francès, 1958] and referred to in Section 2.2.2. However, the results were poor, as

could already be expected by observing the high-frequency non-melodic notes (many of them ghost notes) depicted in Figure 5.3.

Hence, intensity was initially employed as the main cue for figure-ground separation. The most salient notes were selected by deleting non-dominant notes and resolving situations where note overlapping occurs, as described in the following paragraphs.

5.3.1. Elimination of Non-Dominant Notes

A. Removal of Low-Pitch Notes

Before selecting the most salient notes, it is important to take into account that bass sounds are usually very intense. Thus, we first exclude notes in low frequency ranges, where the bass is most likely to be found. This aims to prevent the selection of too many erroneous notes, which would put at risk melody smoothing in the next step.

In this way, the algorithm starts by removing notes below MIDI note number 50 (146.83 Hz), defined in the *minMIDINote* parameter.

Therefore, our approach is biased towards selecting middle and high-frequency notes, which indeed corresponds to most real situations. Anyway, low-frequency notes may still be selected in the next stage, where this restriction will be relaxed, as long as melodic smoothness is ensured.

Alternatively, we could have filtered bass notes in the front-end of the system. However, this would probably lead to the irrecoverable loss of low-frequency melodic notes.

B. Definition of Song Segments and Disposal of Low-Salience Notes

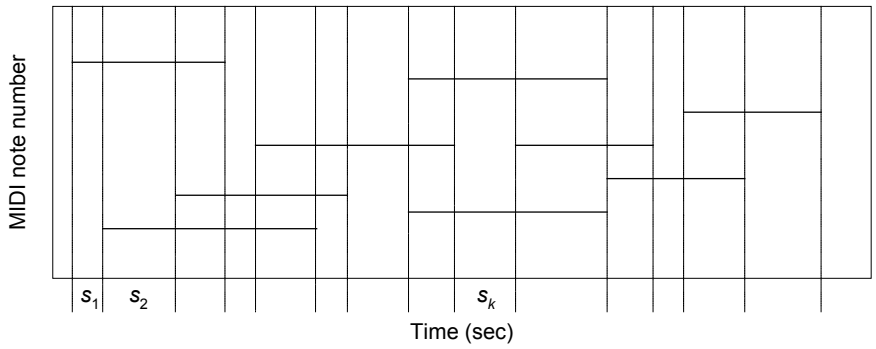


Figure 5.4. Definition of segments in a song excerpt.

At this point, low-salience notes are discarded. To this end, we segment the song excerpt under analysis, as illustrated in Figure 5.4, where, s_k denotes the k^{th} segment.

In each segment, we determine the three most salient notes (set in the *numTop* parameter), based on the average pitch salience of each note in each segment, as in (5.2):

$$\begin{aligned} avgSegmSalience[j,k] &= \frac{\sum_{i \in A_{j,k}} salience[j,i]}{\#A_{j,k}}, \quad k = 1,2,\cdots,numSegm \\ A_{j,k} &= \left\{ \begin{array}{l} i \ : \ i \geq \text{segment start frame and } i \leq \text{segment stop frame} \cdots \\ \text{and } salience[j,i] \neq 0 \end{array} \right\} \end{aligned} \tag{5.2}$$

There, *salience*[*j*, *i*] denotes the salience of the j^{th} note in the i^{th} frame (as computed in Section 3.3.4) and *avgSegmSalience*[*j*, *k*] stands for the average salience of the same note in the k^{th} segment, calculated only in the non-empty frames.

Then, we dispose of all notes that are non-dominant, i.e., that are not in the most salient segment for at least 35% of their total number of frames (set in the *minTopPercDur* parameter) or are not in the three most salient segments for at least 80% of their total duration (defined in the *minNumTopPercDur* parameter).

5.3.2. Resolution of Note Overlaps

After discarding the non-dominant notes, it often happens that the remaining ones overlap in time, which should not be permitted. With the purpose of handling such situations, we first analyze the possible types of time overlapping between pairs of notes.

A. Determination of Overlapping Type

We have identified six overlapping types, illustrated in Figure 5.5. There, the reference note (thick line) is, by definition, the one with the earliest onset time. The other horizontal lines represent time spans of hypothetical notes that overlap in time with the reference note.

The first considered overlapping type corresponds to the situation where the two notes have approximately the same onsets and endings. A maximum allowed distance between the onsets and endings, *maxOnsetDist*, was defined as in the previous section (62.5 msec, which leads to 11 frames and, thus, 63.9 msec in reality).

In the second and third overlapping types, the two notes have common onsets but different endpoints.

In the fourth possibility, the notes under comparison have equal endings but differ-

ent onsets.

Finally, in the fifth and sixth overlapping types, the two notes have neither common onsets nor common endings. The fifth type denotes inclusions, where the second note starts after and ends before the reference note. The sixth type corresponds to the situations where the notes intersect, i.e., the second note starts and finishes after, respectively, the beginning and the ending of the reference note, considering again the maximum allowed difference.

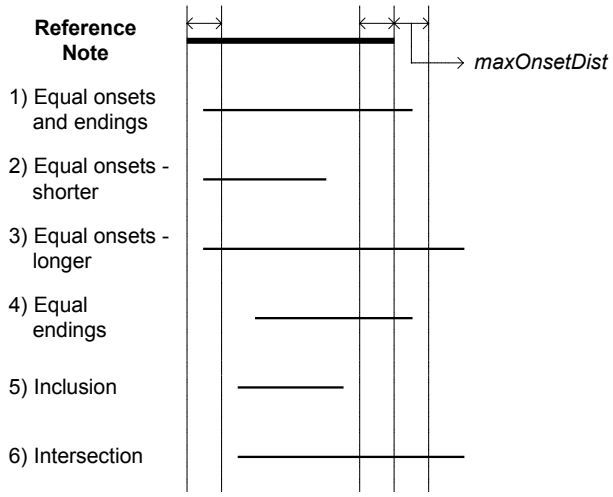


Figure 5.5. Types of note overlapping.

B. Resolution of Overlapping: Elimination or Truncation of Notes

Each candidate note is then compared with the notes that overlap it in time and the overlapping type is determined. In short, preference is given to the note with the highest average salience in their common time interval, causing the elimination of the other note, or to its truncation at the beginning or ending.

For example, in the second overlapping type in Figure 5.5, if the reference note has the highest average salience in the common time interval, the second note is deleted. In the opposite case, the second note is left unchanged whereas the reference note is truncated at its beginning, i.e., it will start later, immediately after the second note ends.

The same reasoning applies to all cases, except for inclusions (situation 5). Here, if the second note is the strongest one, only the beginning or the ending of the reference note is kept, depending on the salience of each of them.

Additionally, the original note timings are saved for future restoration, in case any of

the selected notes is removed in the following stages.

5.3.3. Putting It All Together

The results of the implemented procedures are illustrated in Figure 5.6 for an excerpt from Pachelbel's Kanon. There, the actual melody's notes are gray; the notes selected by the algorithm are black; and dashed lines represent notes that were kept after the elimination of ghosts.

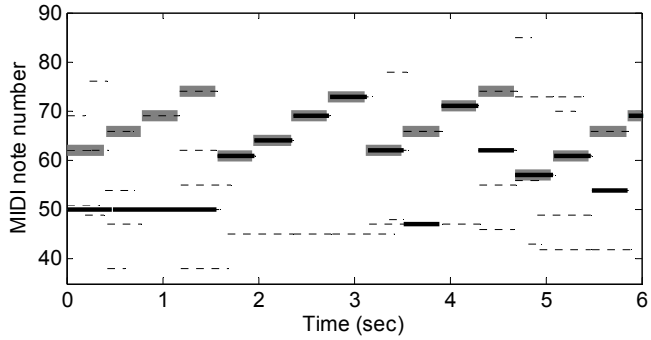


Figure 5.6. Results of the algorithm for selection of the most salient notes.

We can see that some erroneous notes are extracted, whereas true melody notes are excluded. Namely, some octave errors occur. In fact, one of the limitations of only taking into consideration note salience is that the notes comprising the melody are not always the most salient ones. In this situation, wrong notes may be selected as belonging to the melody, whereas true notes are left out. This is particularly clear when abrupt pitch transitions are found, as can be seen in the previous figure. Hence, we improved our method by smoothing out the melody contour, as discussed in the next section.

The process for extraction of salient notes is summarized in Algorithm 5.2. Parameter definition is presented in Table 5.2.

Algorithm 5.2. Selection of the most salient notes.

1. Sort all notes in ascending onset time order.
2. Discard notes i, such that MIDI(i) < 50.
3. Eliminate non-dominant notes.
   3.1. Define song segments, according to Figure 5.4.

3.2. For each segment, *s*,

3.2.1. Compute the average salience of each note in *s*.

3.3. For each note, *i*:

3.3.1. Find the segments where note *i* is the most salient one and calculate the corresponding percentage of frames, *percTop* (comparing to the note's total number of frames).

3.3.2. Compute *percTop3* in a similar way, by considering the segments where note *i* is in the *numTop* = 3 most salient ones.

3.3.3. If *percTop* < *minTopPercDur* (35%) and *percTop3* < *minNumTopPercDur* (80%), delete note *i*.

4. Resolve note overlapping.

4.1. Save the original note beginnings and endings.

4.2. For each resulting note, *i* (reference note):

4.2.1. Look for a note *j* that overlaps note *i*, i.e., onset(*j*) ≤ ending(*i*).

4.2.2. Determine the overlapping type (Figure 5.5).

4.2.3. Calculate the average salience of each note, avgSal, in their common time interval.

4.2.4. If avgSal(*j*) ≤ avgSal(*i*) eliminate or truncate note *j* (based on the overlapping type) and repeat step 4.2.1. until no more notes are found.

4.2.5. Otherwise, delete or truncate note *i* and repeat step 4.2 for the next note.

5. Return the obtained melody notes (salient notes).

Parameter Name	*Parameter Value*
minMIDINote	50
numTop	3
minNumTopPercDur	0.8
minTopPercDur	0.35

Table 5.2. Parameters for extraction of salient notes.

5.4. Melody Smoothing

In an attempt to demonstrate that musicians generally prefer to use smaller note steps, the psychologist Otto Ortmann counted the number of sequential pitch intervals of different sizes in several songs by classical composers. He found out that the smallest ones happen more frequently and that the number of occurrences of each roughly decreases in inverse proportion to the size of the interval [Bregman, 1990, pp. 462]. Wei Chai also presented some statistical results for the music corpora used in her QBH system [Chai, 2001, pp. 46-47]. In the used dataset, comprising mostly folk music but also classical and pop/rock songs, interval histograms were computed which also showed that the frequency of occurrence of sequential intervals dropped as the interval size increases, being just about zero above seven semitones. Interestingly, such drop was not monotonously decreasing. Rather, some intervals were more frequent than others, forming an oscillating pattern that decreased towards zero. For instance, until three semitones, odd intervals (both positive and negative) are clearly less frequent than the even ones. After that, the oscillation continues, but now the even intervals are less frequent.

So being, we improved the melody extraction stage by taking advantage of this melodic smoothness principle. Although this might be a culturally dependent principle, it is relevant at least in Western tonal music, the one considered in this research work. The basic idea is to detect abrupt pitch intervals and replace notes corresponding to sudden movements to different pitch registers by notes that smooth out the extracted melody.

5.4.1. Octave Correction

We start to improve the tentative melody that results from the selection of the most salient notes by performing octave correction. In effect, octave errors might occur because, usually, Algorithm 5.1 does not eliminate all ghost harmonically-related notes.

In order to correct these errors, we select all notes for which no octaves (either above or below) are found and compute the average of their pitches (expressed as MIDI note numbers). Then, we analyze all notes that have octaves with common onsets (some time difference allowed, as before): if the octave is closer to the computed average, the initial note is replaced by the respective octave.

This simple first step already improves the final melody but a few octave errors, as well as abrupt transitions, are still kept, which will be now worked out.

5.4.2. Resolution of Abrupt Note Transitions

In the second step, we smooth out the melodic contour by deleting or replacing notes

corresponding to sudden movements to different pitch registers.

A. Definition of Regions of Smoothness

To this end, we first define regions of smoothness, i.e., regions with no abrupt pitch intervals. Here, intervals above a fifth, i.e., seven semitones (set in the *maxMIDIinterval* parameter), are defined as abrupt, as illustrated in Figure 5.7 for notes a_1, a_2 and a_3 (in bold). The maximum interval was set in conformity with the importance of the perfect fifth in Western music. Other intervals were evaluated as well, but, in the used excerpts, best results were obtained with the fifth. In the example in Figure 5.7, four initial smooth regions are detected (R_1, R_2, R_3 and R_4).

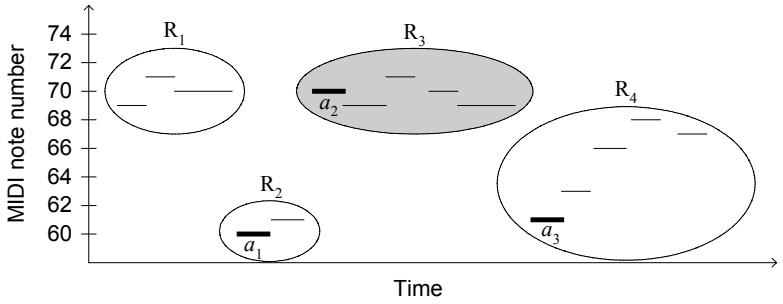


Figure 5.7. Regions of smoothness.

B. Validation of Regions and Analysis of Neighbors: Note Removal or Replacement

We then define the longest region[55] as a correct region (region R_3, in Figure 5.7, filled in gray) and determine the allowed note range for its adjacent regions (R_2 and R_4).

Regarding the region immediately before the longest one (the "left region", R_2), we define its allowed range based on the first note of the correct region, i.e., note a_2 with MIDI note number 70. Keeping in mind the importance of the perfect fifth, the allowed range for the left region is 70 ± 7, i.e., [63, 77]. As region R_2 contains no note in the determined range, this region is a candidate for elimination. However, before deletion, we first look for octaves of each of its notes in the admitted range. In case at least one octave is found, the note in cause is replaced by the respective octave and no note is deleted in this iteration. Otherwise, all the notes in the region are eliminated.

As for the region immediately after (the "right region", R_4), we carry out a similar

[55] The length of each region is calculated as the sum of the lengths of all its notes.

analysis. Hence, we define the allowed range based on the last note of the correct region, e.g., 69 in this example, resulting in the range [62, 76]. Since region R_4 contains a few notes in the derived range, its first note (i.e., note a_3) is marked as non-abrupt and regions R_3 and R_4 are joined together (still, if an octave of note a_3 is found in the allowed range, this octave is used instead of the initial note). In this way, abrupt transitions are permitted in case adjacent regions have notes in similar ranges. This situation occurs in some musical pieces as, for example, Pachelbel's Kanon (Figure 5.6 and Figure 5.8).

If no notes are replaced or eliminated in the current region, the remaining regions are similarly analyzed, in descending length order. If no change at all is done in all regions, the algorithm stops. Otherwise, whenever a change is accomplished, the set of operations for definition of regions of smoothness, analysis of neighbors and deletion/substitution is repeated until no change is done. In the successive iterations, regions of smoothness are defined taking into consideration the notes previously marked as non-abrupt, e.g., note a_3 in region R_4 in the above descriptions. Therefore, in the next iteration, regions R_3 and R_4 will be joined into one region.

5.4.3. Gap Filling

As a result of region elimination, the respective notes need to be replaced by other notes that are more likely to belong to the melody, according to the smoothness principle.

Thus, we fill in each gap with the most salient notes that start in that time interval and are in the allowed range. Again, we do not permit note overlapping. In this gap filling scheme, the previous restriction on the minimum permitted pitch (in the selection of the most salient notes) no longer applies: the most salient note in the allowed range is selected, regardless of having a low MIDI note number. Indeed, that constraint was imposed as a necessity to prevent the selection of too many erroneous notes (particularly bass notes), which would jeopardize melody smoothing. Hence, we kept the general assumption that melodies are contained in middle frequency ranges, but admitting now the selection of lower-pitch notes, as long as the smoothness requirement is fulfilled.

However, because of gap-filling, accompaniment notes may be inadvertently added, which we make an effort to discard in the next stages of the algorithm.

5.4.4. Note Timing Restoration

Finally, due to note elimination and substitution, previously truncated notes may now be restored to their original temporal intervals (or at least partly extended). In this way, if any of the frames corresponding to the initial interval of a truncated note become empty, its start and endpoints are tentatively reset to their original values. If this is not possible,

they are adjusted to the position of the first empty frame after the note immediately before (onset) or to the position of the last empty frame before the following note (ending).

5.4.5. Putting It All Together

The results of the executed procedures are illustrated in Figure 5.8 for the same excerpt from Pachelbel's Kanon presented before. We can see that only one erroneous note was output (signaled by an ellipse), corresponding to an octave error. This example is particularly challenging to our melody-smoothing approach as a consequence of the periodic abrupt transitions present. Yet, the performance was quite good.

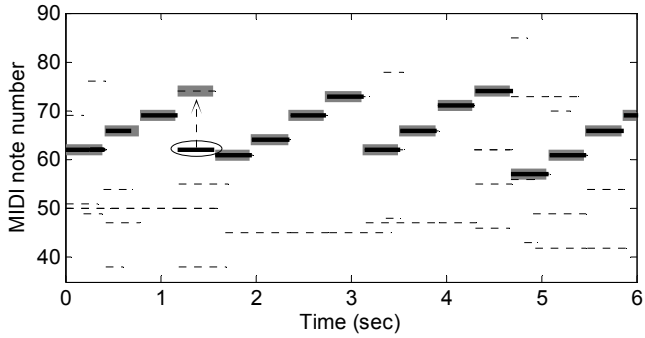


Figure 5.8. Results of the melody-smoothing algorithm.

The described implementation of the melodic smoothness principle is summarized in Algorithm 5.3. Parameter definition is presented in Table 5.3.

Algorithm 5.3. Melody extraction using melodic smoothness.

1. Correct octave errors:
   1.1. Select all notes with no octaves (above or below).
   1.2. Compute the average of their pitches, *avgMIDI*.
   1.3. For each note, *i*:
      1.3.1. Look for a note *j*, such that:
         a) (|onset(*i*) – onset(*j*)| ≤ *maxOnsetDist* and
         b) MIDI(*i*) – MIDI(*j*)| = 12*k*.
      1.3.2. Replace the current note, *i*, by the found note

(the octave), *j*, if *j* is closer to *avgMIDI*, i.e., if |MIDI(*j*) – *avgMIDI*| < |MIDI(*i*) – *avgMIDI*|.

2. Smooth out the melodic contour by deleting or replacing notes corresponding to sudden movements to other pitch registers:
   2.1. Define regions of smoothness, which are delimited by abrupt notes, i.e., notes *i* such that |MIDI(*i*) – MIDI(*i*-1)| > 7 (see Figure 5.7).
   2.2. Select the longest region, *r*, as a correct region.
   2.3. Analyze the regions immediately before and after region *r* for possible note deletion or octave substitution.
   2.4. If any deletions or substitutions are performed in step 2.3, repeat from step 2.1; otherwise, repeat from step 2.2 for the next longest region, until all regions are analyzed.
3. Repeat step 2 until no deletions or substitutions are accomplished (take into consideration notes marked as non-abrupt).
4. Fill in gaps:
   4.1. Look for empty intervals, *g*, such that duration(*g*) > *minNoteLen* (125 msec).
   4.2. For each interval, *g*:
      4.2.1. Look for a set of notes, *i*, such that:
         a) onset(*i*) occurs during *g* and
         b) MIDI(*i*) is in {MIDI(last note before gap) ± 7} or is in {MIDI(first note after gap) ± 7}.
      4.2.2. Select the most salient notes in gap *g*, as in Algorithm 5.2 (except that step 2 is not run).
5. Restore the truncated notes to their original time intervals, as much as possible.
6. Return the obtained melody notes (salient notes that satisfy the melodic smoothness principle).

Parameter Name	*Parameter Value*
maxMIDIinterval	7

Table 5.3. Melody smoothing parameters.

5.5. Elimination of Spurious Accompaniment Notes

In the previously described stages of melody extraction, the algorithm outputs the most salient notes at each time in the allowed note range. Consequently, false positives may turn up. Such notes may be output both when pauses between melody notes are sufficiently long (giving rise to spurious accompaniment notes that are added in between) and when the solo is quiet (e.g., the solo has stopped and another instrument takes the lead for some time). Hence, false positives should be eliminated, both regarding spurious notes and notes that dominate when the solo is absent.

The segregation of melodic information in the mixture is in fact an important aspect in any melody extraction task. In this section, we described the efforts conducted for the removal of spurious notes. Note clustering will be described in the next section.

We observed that, usually, spurious accompaniment notes have lower saliences and shorter durations, giving rise to clear intervals with low note salience and length. In this way, we attempt to dispose of false positives by detecting and discarding the notes in those regions.

5.5.1. Analysis of the Salience Contour

Regarding the salience contour, we start by computing the average salience of each note in the extracted melody and then looking for deep valleys in the note sequence.

As with salience-based track segmentation, we detect deep minima in the salience contour (see description on page 137 of Section 4.4.1.B). Here, we define a valley as being profound if its prominence is at least 30 units, a value set in the *minDeepValleyProm* parameter (recall that saliences were normalized to the [0; 100] in the pitch detection stage). Hence, notes in deep valleys of the salience contour are discarded.

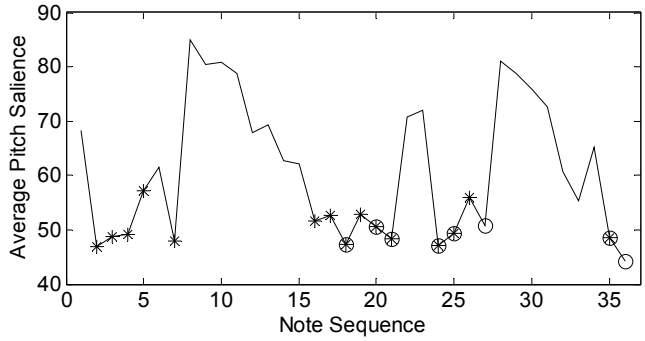


Figure 5.9. Pitch salience contour (jazz3 excerpt).

However, low-salience notes that did not correspond to local minima are not removed. Therefore, the procedure for detection and elimination of clear minima is executed iteratively until no deletions occur. Finally, previously truncated notes are restored to their original time intervals, as in Section 5.4.4.

A jazz excerpt (jazz3 sample in Table 2.1), where the solo is often absent, was chosen to illustrate the conducted operations. The note-salience contour of the employed sample is depicted in Figure 5.9, where '*' denote false positives and 'o' represent deleted notes. It can be seen that two true notes were nevertheless removed. Besides, with a lower elimination threshold, a few more false positives would have been deleted, but best overall results were attained with the defined threshold.

5.5.2. Analysis of the Duration Contour

As for the duration contour, we proceeded likewise. However, we observed that duration variations are much more common than salience variations. This was expected since tone lengths tend to vary considerably.

In this way, we decided to eliminate only isolated abrupt duration transitions, i.e., individual notes whose adjacent notes are significantly longer. Here, we define a note as being too short if its duration is less than 20% the one of the smallest of its neighbors (a value set in the *minPercDur* parameter). Additionally, in order not to inadvertently delete short ornamental notes, corresponding to frequently used whole-step grace notes, a minimum difference of three semi-tones was defined (*maxSemiToneDiff* parameter).

5.5.3. Putting It All Together

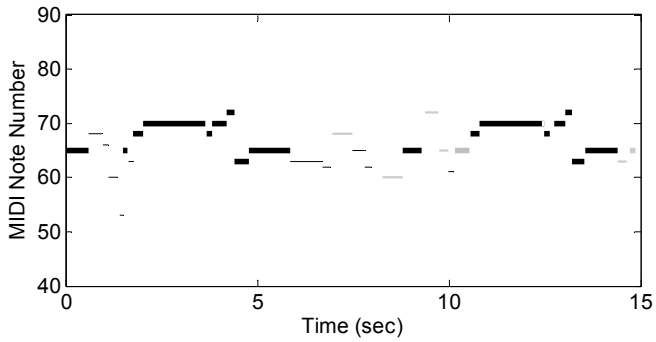


Figure 5.10. Results of the algorithm for elimination of spurious notes.

The melody notes extracted after analysis of the pitch salience and duration contours are visualized in Figure 5.10. There, thick black lines denote true positives, thin black lines represent false positives, thick gray lines denote deleted melodic notes and thin gray lines represent deleted non-melodic notes. It can be seen that, even though a few extra notes are disposed of (including two true melodic notes), some false positives remain. In this way, we carried out a pilot study aiming to further discriminate between melodic and accompaniment notes, described in the next section.

The algorithm for elimination of spurious notes is summarized in Algorithm 5.4. Parameter definition is presented in Table 5.4.

Algorithm 5.4. Elimination of spurious notes.

1. Delete notes in lower salience intervals:
   1.1. Obtain the pitch salience contour:
      1.1.1. Calculate the average pitch salience of each note, i, avgSal(i).
      1.1.2. Form the salience contour as the sequence of average note saliences.
   1.2. Remove notes in lower salience regions of the salience contour:
      1.2.1. Detect all clear minima, as in salience-based track segmentation (see description on page 137 of Section 4.4.1.B).
      1.2.2. Repeat 1.2.1 until no more notes are deleted.
2. Delete notes corresponding to abrupt duration decreases:
   2.1. Define the duration contour (similarly to step 1.1).
   2.2. Delete isolated notes corresponding to deep valleys in the duration contour:
      2.2.1. Detect all local minima in the contour.
      2.2.2. For each local minimum, i:
         a) If duration(i) < *minPercDur*·duration(i+1) (i.e., 0.2·duration(i+1)) and duration(i) < *minPercDur*·duration(i-1) and |MIDI(i) – MIDI(i+1)| ≥ *maxSemiToneDiff* (i.e., 3) and |MIDI(i) – MIDI(i-1)| ≥ *maxSemiToneDiff*, delete note i.

```
    2.3. Restore truncated notes to their original time inter-
         vals, as much as possible.
3. Return the resulting melody notes.
```

Parameter Name	*Parameter Value*
minDeepValleyProm	30
minPercDur	0.2
maxSemiToneDiff	3

Table 5.4. Parameters for elimination of spurious notes.

5.6. Note Clustering

As observed in the previous section, notes from the most prominent accompaniment are usually output when the solo stops. It can be argued that this corresponds to the way humans memorize songs: a continuous "line" that comprises both melody per se and major accompaniments. However, since our goal is to extract the melody in a strict sense (not a predominant pitch line), the accompaniment should be eliminated. To this end, we attempt to discriminate true notes from false positives via note clustering.

This work is related to the classification of musical instruments in polyphonic contexts. Only little work has been conducted in this field (e.g., [Kitahara *et al.*, 2005; Vincent and Rodet, 2004; Eggink and Brown, 2003]), with limited results so far. In fact, this is a complex task since, in one hand, it is difficult to define acoustic invariants that are good timbre correlates and, on the other hand, the proposed features are hard to measure in a polyphonic environment due to spectral overlapping between sources.

We start by extracting a set of acoustic features related to timbre, using them as a basis for note source discrimination. The dimensionality of the feature space is reduced with recourse to Principal Component Analysis (PCA) and the best set of features is iteratively chosen via forward selection. Finally, clustering is implemented with Gaussian Mixture Models, where true notes and false positives are separated (similarly to [Marolt, 2004]).

For comparison purposes, we also performed note clustering on the entire set that results after ghost note elimination. This is carried out before the selection of salient notes, melody smoothing and deletion of spurious notes.

5.6.1. Acoustical Correlates of Timbre

The concept of timbre is quite vague. In effect, it has been defined (if we can say so...) as "the quality of a sound by which a listener can tell that two sounds of the same loudness and pitch are dissimilar" [ANSI, 1973]. Here, timbre is described by exclusion: rather than explaining what timbre is, it is said what timbre is not. Moreover, this definition appears incomplete, since it does not seem to leave room for unpitched sounds.

Timbre is a complex and broad perceptual attribute of sound. Basically, it answers the question to "what something sounds like" [Bregman, 1990, pp. 93] and is intimately related to the identification of sound sources: different instrument types sound differently, i.e., have different timbres. The biggest difficulty concerning its automatic processing and analysis is that timbre is not explained by a single acoustic property.

Often, timbre is qualitatively described using terms such as "bright", "dull", "scratchy" or "shimmering" [Wold *et al.,* 1996]. However, the quantitative side of the subject is much more intricate. Indeed, the extraction of physical features that are good timbre correlates is a difficult issue, for which no definitive answers are available thus far.

It is well believed that, at least for isolated tones, the perception of timbre is influenced by the frequency content of the signal at steady-state, namely spectral centroid and the relative amplitudes of harmonic components, as well as the signal's temporal envelope and the temporal behavior of the harmonics, in which the attack transient assumes particular importance. For example, the spectral centroid is related to the perceived brightness of a sound whereas the temporal evolution of the harmonics (e.g., the differences in the respective onset times) is associated with the roughness of the sound. However, these features change dramatically from note to note across the playing range of an instrument. Yet, the same timbre is heard [Handel, 1989, pp. 170].

Other features such as inharmonicity (introduced in Section 3.1.1) also give instruments a characteristic richness. In reality, electronic instruments with exact harmonic frequencies sound cold and artificial.

Intensity also has an effect on timbre. For example, sounds become more piercing at higher intensities [Handel, 1989, pp. 168]. In fact, in such cases, the excitation contains a greater amount of higher harmonics, and so the relative amplitudes of harmonic components are modified. Furthermore, in musical instruments each harmonic has its own temporal envelope, which is also affected by intensity, as well as frequency.

Resonant frequencies might look more appealing for timbre analysis. Nevertheless, neither the relative nor the absolute frequencies and intensities of the formants of musical instruments or the human voice are invariant timbre cues. Indeed, with respect to the human voice, the formant frequencies depend on aspects such as genre or age. As for musical instruments, these are influenced, for example, by the quality of the material used by the manufacturer [Handel, 1989, pp. 120, 172].

Moreover, timbre perception is also affect by the ongoing context, e.g., tone frequency, intensity and duration, as well as previous knowledge regarding the sounds of musical instruments and performing styles. The human auditory system makes extensive use of this kind of data during the listening experience.

To conclude, "timbre is the result of many changing and interacting acoustic properties" [Handel, 1989, pp. 173]. Thus, we will attempt to model timbre with recourse to different sorts of acoustic features in the extracted musical notes.

5.6.2. Feature Extraction

Feature extraction consists on the computation of numerical quantities able to represent, in a condensed and meaningful way, relevant information that might be hidden in a raw data set.

Most of the sound processing features suggested in the literature have been defined in the context of speech signal analysis, e.g., in tasks such as compression, telephony, speech recognition and synthesis [Tzanetakis, 2002, pp. 26]. Here, two representations have deserved particular attention: Linear Prediction Coefficients and Mel-Frequency Cepstral Coefficients.

By the end of the 1990's, when research on sound processing was becoming more diversified, features for representing non-speech signals started being investigated. Particularly, musical instrument recognition and genre classification gave a strong impulse towards the definition of musical-content features [Kitahara *et al.*, 2005; Vincent and Rodet, 2004; Eggink and Brown, 2003; Tzanetakis, 2002; Eronen, 2001; Agostini *et al.*, 2001; Martin, 1999; Fujinaga, 1998].

In our work, we are particularly interested in determining the source of each note to the extent that melodic and non-melodic notes might be discriminated. Our goal is partly connected to musical instrument identification. This is often carried out on a note-by-note basis, which corresponds to our case. However, most studies on instrument recognition are conducted in monophonic contexts, where clean and isolated tones are available. This is certainly not our situation, since mixtures of simultaneous notes are the common condition. Only little work has been devoted to instrument recognition in polyphonic audio (e.g., [Eggink and Brown, 2003]), so far with limited accuracy.

In any case, we make use of several features that have been proposed in the literature with that purpose. Such features aim to capture pitch, intensity and timbre content using both the attack and steady-state regions of each note.

Particularly, spectral shape (e.g., spectral centroid or skewness), attack transient (e.g., frequency slope) and duration, intensity and pitch-related features are often employed. In order to extract such features, especially, the ones associated with spectral shape, the

frequencies and magnitudes of tone harmonics must be acquired. This is described in the following paragraphs.

A. Front-End for Feature Calculation

The collection of features utilized in this work (listed on page 181) was extracted on top of the auditory front-end used in the pitch detection stage. Thus, the harmonic frequencies and magnitudes of each pitch candidate in each frame were obtained directly from a correlogram frame, by using the respective correlogram columns, as illustrated in Figure 5.11 (see also Figure 3.14c, in Section 3.3.5). There, the correlogram column corresponding to a time lag of 5.4 msec (signaled with a black vertical line) is expanded in Figure 5.11b. Although we used a linear frequency axis for ease of harmonic visualization, the center frequencies pertaining to each frequency channel are distributed along a logarithmic scale, as described in Section 3.3.1.

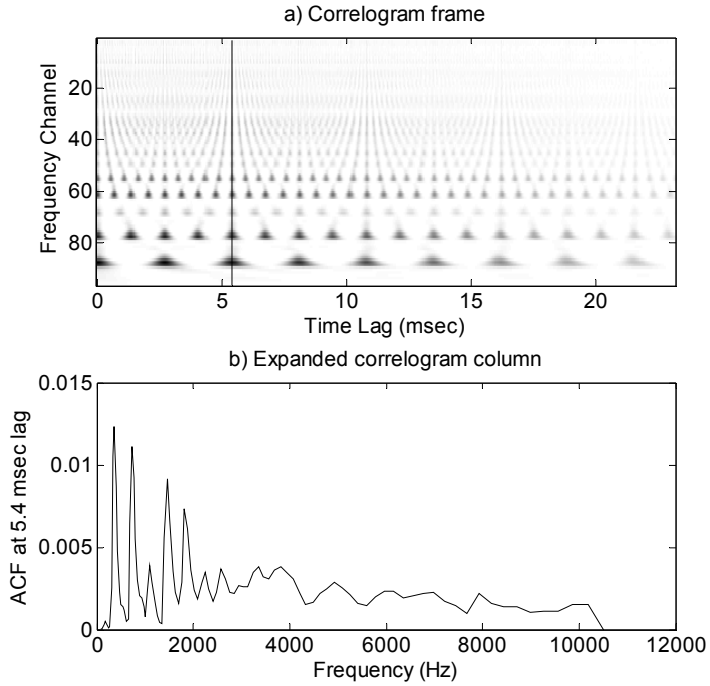


Figure 5.11. Detection of harmonics from the correlogram.

Matching of Peaks in the F0 Column

Then, for the column corresponding to the estimated pitch period, local peaks are

detected and matched to the expected theoretical frequencies of each harmonic (including the first one).

If no peak is found in the allowed range of the target frequency partial, which we have defined as a maximum distance of *harmonicCentsRange* = 40 cents from its theoretical value, the filterbank channel whose center frequency is closest to it is selected. Hence, in this case the harmonic frequencies and magnitudes are the ones of the filter channel. This is implemented much in the same way as Martin did [Martin, 1999, pp. 69-84].

In terms of the maximum number of harmonics, different values were experimented and best results were achieved with exactly six. Therefore, only *numHarmonics* = 6 frequency partials are used.

Determination of the Steady-State Region

Two types of features are then computed, according to the appropriate note region where they should be acquired (i.e., attack or steady-state). Thus, we first determine the stability regions of each note.

Based on the median frequency of the current note (as obtained in Section 4.3), we define the start of the steady-state region as the first frame of the pitch track where the frequency is less than *maxCentsDistMedian* = 50 cents apart from the median. As for its ending, it will correspond to the last point where the frequency is less than *maxCentsDistMedian* apart.

This strategy is not optimal since a few frames where the frequency is still evolving are inadvertently utilized in steady-state feature computation. Nonetheless, it copes well with notes with strong vibrato, where the steady region is not always easily identified.

The steady-state region is the preferred time interval for calculating spectral features (e.g., spectral centroid). In effect, as a consequence of the transient noise present during note attacks, these features only make use of the frames in the steady part of the signal. In this way, their values are evaluated with recourse to the harmonic frequencies and magnitudes in all frames of the steady-state region. Regarding note attacks, these are the home of temporal features (e.g., frequency slope, onset duration).

B. Used Features

In order to obtain information on the source of each note, we employ a set of features that might be able to capture pitch, intensity and timbre content, using both the attack and steady-state parts of each note. The following were computed, complying with the described criteria. We start the description with steady-state features.

Harmonic Frequency

The exact frequency values of each harmonic provide information on the harmonic-

ity of the note under analysis. Also, this feature is the basis for other harmonicity-related features. Harmonic frequencies are directly derived from the described front-end, except that only the steady part of the note is used, due to increased harmonic stability.

Relative Harmonic Frequency Ratio

The same as before, except that now relative values are used. Formally, it turns out (5.3). There, *hfr*[*i*, *k*] denotes the harmonic frequency ratio regarding the k^{th} harmonic (in a total of *NH*) in the i^{th} frame of the steady-state region. In the same expression, f_H represents harmonic frequency.

$$hfr[i,k] = \frac{f_H[i,k] - f_H[i,k-1]}{f_H[i,1]} \quad , \quad \begin{matrix} i = start_{ss}, \cdots, end_{SS} - 1 \\ k = 2, 3, \cdots, NH \end{matrix} \tag{5.3}$$

Basically, *hfr* is the ratio of the difference between the frequencies of the current harmonic, *k*, and the previous one, over the first harmonic. In case of perfect harmonicity, i.e., if all harmonic frequencies conform to their theoretical values, *hfr* is always one.

Spectral Inharmonicity

Another harmonicity-related feature, as the name suggests. This feature is calculated as the cumulative sum of differences of each harmonic frequency from its theoretical value [Agostini *et al.*, 2001], as in (5.4). There, the distances are measured in cents. Again, the steady-state part of the note is used.

$$\begin{aligned} inharmonicity[i] &= \sum_{k=2}^{NH} \left| diff[i,k] \right| \\ diff[i,k] &= f_H^{cent}[i,k] - \left(f_H^{cent}[i,1] + 1200 \cdot \log_2(k) \right) \end{aligned} \quad , \quad i = start_{ss}, \cdots, end_{SS} \tag{5.4}$$

Harmonic Magnitude

The magnitudes of note harmonics give important information regarding timbre. Also, this is the basis of other spectral-shape features. Harmonic magnitudes are directly picked up from the front-end, resorting only to the steady-state region of the note.

Relative Harmonic Magnitude Ratio

The same as before, except that now relative values are used (based on [Martin, 1999, pp. 90]). Formally, it comes (5.5):

$$hmr[i,k] = \frac{M_H[i,k]}{M_H[i,k-1]} \quad , \quad \begin{matrix} i = start_{ss}, \cdots, end_{SS} - 1 \\ k = 2, \cdots, NH \end{matrix} \tag{5.5}$$

In short, *hmr* is defined as the ratio of the magnitude of the current harmonic, $M_H[k]$, over the magnitude of the previous one.

Spectral Centroid

This is a simple yet important feature in the characterization of instrument timbre. Indeed, besides being a measure of spectral shape, it also correlates well with the perceived sound brightness, e.g., higher values correspond to "brighter" sounds, i.e., sounds with higher high-frequency content. Formally, the spectral centroid is defined as the first moment of the magnitude spectrum with respect to the frequency, i.e., the center of gravity of the magnitude spectrum, as follows (5.6) (e.g., [Tzanetakis, 2002, pp. 32]):

$$f_{SC}[i] = \frac{\sum_{k=1}^{NH} M_H[i,k] \cdot f_H[i,k]}{\sum_{k=1}^{NH} M_H[i,k]}, \quad i = start_{ss}, \cdots, end_{SS} \tag{5.6}$$

Relative Spectral Centroid

This feature aims to provide a fundamental-frequency-free indication of the brightness of a given harmonic sound. In this way, the relative spectral centroid, f_{RSC}, is simply computed as the ratio of the spectral centroid over the sound's F0, according to (5.7) [Martin, 1999, pp. 87]. Once more, this feature is calculated in the steady-state region.

$$f_{RSC}[i] = \frac{f_{SC}[i]}{f_0[i]}, \quad i = start_{ss}, \cdots, end_{SS} \tag{5.7}$$

Spectral Skewness

Spectral skewness [Agostini *et al.*, 2001] is another measure of spectral shape, evaluated as the sum of harmonic magnitudes, weighted by their respective inharmonicities, according to (5.8). There, *diff* is defined as in (5.4).

$$skewness[i] = \sum_{k=1}^{NH} M_H[i,k] \cdot \left| diff[i,k] \right|, \qquad i = start_{ss}, \cdots, end_{SS} \tag{5.8}$$

Spectral Irregularity

Still another spectral-shape feature, irregularity measures the amount of local spectral change. It corresponds to the standard deviation of time-averaged harmonic amplitudes from a spectral envelope [Eronen, 2001, pp. 38], as in(5.9):

$$irregularity[i] = \sum_{k=2}^{NH-1} M_H[i,k] - \frac{M_H[i,k-1] + M_H[i,k] + M_H[i,k+1]}{3}, \quad i = start_{ss}, \cdots, end_{SS} \tag{5.9}$$

Attack Duration

Besides spectral-shape features, attack transient features are also calculated, given their importance in the perception of timbre.

Namely, note attack duration correlates well with the type of coupling between the excitation and resonant structures (e.g., short attacks indicate tight coupling) [Eronen, 2001, pp. 34]. This is computed as the time interval between onset time and the start of the steady-state region.

Attack Energy Slope

The attack energy slope [Martin, 1999, pp. 86] is hard to evaluate for notes with many missing values in the attack. Thus, its calculation was simplified by interpolating the first and last salience values at the beginning of the note (i.e., until the start of steady-state).

Harmonic Onset Time Delay

Onset asynchrony is also an important cue for timbre perception. Hence, harmonic onset times are measured as the absolute time delay of each harmonic compared to the note onset, in this manner (5.10) (based on Martin, 1999, pp. 100):

$$onsetTimeDelay[k] = \left| onsetTime(k) - onsetTime(1) \right|, \quad k = 1,2,\cdots NH \tag{5.10}$$

Salience

This feature is strongly correlated to the intensity of the sound. It is computed in the whole note duration, as in the previous sections.

Pitch stability

Here, we measure the frequency variation over successive time frames [Marolt, 2004]. This feature provides information pertaining to aspects such as pitch jitter or modulation. It is calculated in the steady-state region of the note, as in (5.11):

$$stability[i] = f_0[i+1] - f_0[i], \quad i = start_{ss}, \cdots, end_{SS} - 1 \tag{5.11}$$

Note Duration

The name is self-explaining: this feature represents the total note duration.

C. Computation of Statistical Summaries and Normalization

Finally, rather than storing sequences of feature values, we obtain statistical summaries for each feature. Namely, mean and standard deviation are used, except for those features that have a sole value in each note (e.g., frequency slope, duration).

In addition, each feature vector is normalized to the [0; 1] interval to avoid numerical problems in the subsequent analysis, due to disparate feature ranges.

D. Remarks on Feature Calculation

The computation of some of the features was problematic, as a consequence of the polyphonic context we are working in. Namely, the frequency slope was difficult to calculate for notes with many missing frequency values in the attack. Therefore, the slope was simply measured by interpolating the first and last frequency values at the beginning of the note (i.e., until the start of steady-state). Also, some harmonic magnitudes may be corrupted because of spectral collisions. In this way, those elements should be discarded and clustering should be accomplished following a missing feature strategy [Eggink and Brown, 2003]. This question will be addressed in future developments.

5.6.3. Feature Selection and Dimensionality Reduction

The quantity of implemented features is very high compared to the number of notes available in each song excerpt. Moreover, a high number of features may give rise to the so-called *curse of dimensionality* issue [Bishop, 1995]. Hence, feature selection and dimensionality reduction were carried out prior to clustering.

A. Feature Selection

As previously referred to, it is important to select the best combination of features to enclose. Since it is impractical to analyze every different combination, forward selection was conducted [Bishop, 1995]. Thus, starting from an empty feature set, the algorithm adds, step by step, the feature that leads to the best model accuracy. The combination of features that originates the highest overall performance is then selected.

B. Dimensionality Reduction

Furthermore, the dimension of the feature space is reduced with recourse to Princi-

pal Component Analysis [Bishop, 1995]. This is a widely used technique, whose basic idea is to project the computed feature matrix into a basis that best expresses the original data set, guaranteeing that the variance of the data is best preserved.

The model is founded on four fundamental assumptions: i) linearity; ii) the extracted features follow Gaussian distributions; iii) large variances have important dynamics (i.e., the components of the new basis with higher associated variances represent important dynamics, being therefore designated as principal components); and iv) the principal components are orthogonal. Hence, we assumed our problem to be linear (although extensions to PCA for the non-linear case are available), and that the implemented features conform to a Gaussian distribution.

Given the previous assumptions, the object of PCA can be concisely resumed to finding a projection matrix P whose lines correspond to new basis vectors such that the projected data, Y, is de-correlated. Formally, it comes (5.12):

$$\underset{m\times n}{Y} = \underset{m\times m}{P} \cdot \underset{m\times n}{X}, \qquad \text{such that } R_Y = \frac{1}{n-1} YY^T \text{ is diagonal} \tag{5.12}$$

In (5.12), X denotes the original multi-dimensional data, i.e., our initial feature matrix, m is the number of features, n stands for the number of measurements (i.e., the number of notes) and R_Y is the covariance matrix of the transformed data.

The previous problem is solved by defining the projection matrix P such that its rows are the eigenvectors of XX^T. In this way, each line of P corresponds to a basis vector and the i^{th} element of R_Y stands for the variance of X along the i^{th} principal component.

Finally, we performed dimensionality reduction by keeping the "best" components, i.e., the ones for which the variance is the highest (the best directions of projection) and whose sum amounts for *percVariance* = 90% of the total variance.

Regarding implementation, we made use of the PCA Matlab code provided in the Netlab toolbox [Nabney and Bishop, 1996].

5.6.4. Clustering

Finally, after feature extraction, selection and dimensionality reduction, true melody notes and false positives are discriminated via clustering.

Clustering is a very broad research theme, which could constitute a thesis in itself. In this section, we will only offer a brief overview of the subject, as it applies to our problem of identification of melodic notes. A detailed discussion of clustering techniques, such as neural networks, *k*-means clustering, GMMs, etc., can be found in [Bishop, 1995]. In short, such techniques consist on the automatic grouping of example feature vectors into a set of classes, in an unsupervised fashion.

A. Cluster Definition with GMMs

In our approach, we employ GMMs, which are extensively used for unsupervised data clustering [Bishop, 1995]. Its essential idea is to fit Gaussian distributions to the observed data. Thus, GMMs model the probability density of the observed features by multivariate Gaussian mixture densities, as in (5.13):

$$p(\mathrm{y}\mid\theta)=\sum_{i=1}^{numClusters} w_i p_i(\mathrm{y}\mid\theta_i) \qquad \sum_{i=1}^{numClusters} w_i = 1 \tag{5.13}$$

There, *numClusters* is the number of defined clusters. In order to separate false positives from true melody notes, we defined only two clusters (a *melody cluster* and a *garbage cluster*). In the same expression, p is the PDF of the mixture, y represents a feature vector with dimension m, θ is the full set of parameters and w_i is the weight associated with the i^{th} multivariate Gaussian, p_i, with parameters θ_i, defined as follows (5.14):

$$p_i(\mathrm{y}\mid\theta_i)=\frac{1}{(2\pi)^{m/2}\cdot|\mathrm{R}_i|^{1/2}}e^{-\frac{1}{2}(\mathrm{y}-\mu_i)^{\mathrm{T}}\mathrm{R}_i^{-1}(\mathrm{y}-\mu_i)} \tag{5.14}$$

In (5.14), μ_i and R_i represent, respectively, the mean and the covariance matrix of the i^{th} Gaussian. The complete set of parameters is formally defined as (5.15):

$$\theta=\{\mu_i,\mathrm{R}_i,w_i\},\quad i=1,2,\cdots,numClusters \tag{5.15}$$

B. Parameter Optimization

Our goal is, then, to find the maximum likelihood estimation of θ. This is obtained by maximizing the log-likelihood, as in (5.16).

$$\theta_{ML}=\arg\max_{\theta}\left(\log p(\mathrm{Y}\mid\theta)\right) \qquad \log p(\mathrm{Y}\mid\theta)=\log\prod_{k=1}^{n}p(\mathrm{y}_k\mid\theta) \tag{5.16}$$

There, Y is the feature matrix composed by n feature vectors y_k. This is accomplished iteratively with recourse to the expectation-maximization algorithm [Bishop, 1995], using once again the source code from the Netlab toolbox.

Before optimization, the parameters must receive adequate initial values. In terms of the centers, these are initialized with the *k*-means clustering algorithm. As for covariance

matrices, these are typically assumed diagonal, for efficiency reasons. This presumes feature independence, which is the case since, after PCA, the projected features are de-correlated. Moreover, the diagonals start with unity values. Finally, the mixture weights are uniformly initialized.

Optimization is then iteratively conducted until a maximum of *maxIter* = 1000 iterations is reached or the difference in the log likelihood between two consecutive iterations is below *minLogDiff* = 0.0001.

C. Note Assignment and Identification of the Melodic Cluster

After optimization, the probabilities of each sample (i.e., note) calculated in each Gaussian, p_i, i.e., the class posterior probabilities, are evaluated. Subsequently, each note is allotted to the cluster where its probability is maximum.

Finally, the melody is assigned to the cluster with maximum salience, where cluster salience is computed as the sum of the average pitch salience of each note multiplied by its duration, as follows (5.17):

$$clustSal[i] = \sum_{k=\text{notes in cluster } i} avgSalience[k,i] \cdot duration[k,i], \quad i = 1,2,\cdots,numClusters$$

$$avgSalience[k,i] = \frac{\sum_{j \in A_k} salience[k,j]}{\#A_{k,i}} \tag{5.17}$$

$$A_{k,i} = \left\{ \begin{array}{l} j \ : \ j \geq \text{note start frame and } j \leq \text{note stop frame} \cdots \\ \text{and } salience[k,j] \neq 0 \end{array} \right\}$$

There, *clustSal*[*i*] denotes the salience of the i^{th} cluster, *avgSalience*[*k*, *i*] and *duration*[*k*, *i*] are, respectively, the average pitch salience and duration of the k^{th} note in that cluster. The average salience of note *k* is only calculated in the set of non-empty frames $A_{k,i}$. In the same expression, *salience*[*k*, *j*] is the salience of the k^{th} note in its j^{th} frame.

D. Clustering on the Whole Note Set

As an experiment, we also investigated a different strategy, where clustering was carried out on the whole note set (the one obtained after ghost note elimination).

Some constraints should be imposed on the performed clustering (e.g., no overlapping between notes) [Marolt, 2004]. Nonetheless, we ignored this issue since the procedures for detection of salient notes and melody smoothing ensure the consistency of the results. Furthermore, harmonically-related note candidates may actually correspond to components of the same note, and so such restrictions would be problematic in this case.

Therefore, notes were clustered with the GMM algorithm, using now five clusters[56]. Then, for each cluster, salient notes were detected, melody smoothing was accomplished and spurious notes were removed.

Finally, the melody was assigned to the cluster with the highest salience, as before.

5.6.5. Putting it All Together

The results for note clustering are illustrated in Figure 5.12, for the same jazz excerpt used before (see Figure 5.10, on page 175). As can be seen, all false positives were eliminated. However, two more melody notes (besides the other two already deleted when possible spurious notes were inspected) were inadvertently discarded.

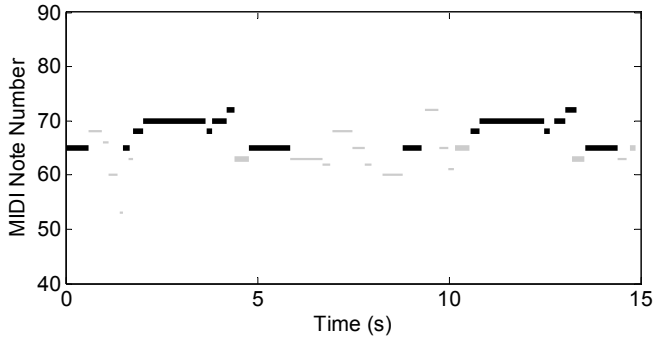


Figure 5.12. Results of the note clustering algorithm (jazz3 excerpt).

The note clustering process is summarized in Algorithm 5.5. Parameter definition is presented in Table 5.5 (on page 191).

Algorithm 5.5. Note clustering.

```
1. Get harmonic frequencies and magnitudes based on the audi-
   tory-model front-end:
   1.1. For each periodicity candidate:
        1.1.1. Select the corresponding correlogram column.
```

[56] We set this parameter by assuming a maximum of five simultaneous harmonic instruments, which seems reasonable to us.

1.1.2. Detect all peaks in the column.

1.1.3. Match peaks to the expected values of each harmonic or select the frequency and magnitude of the closest channel, in a maximum of *numHarmonics*.

2. Determine the steady-state region.

3. Perform feature calculation:

3.1. Measure steady-state features.

3.2. Evaluate attack transient features.

3.3. Compute feature means and standard deviations for the frame-based features.

3.4. Normalize features to the [0, 1] interval.

3.5. Create the feature matrix:

3.5.1. Assemble each feature vector (column containing the calculated features – means and standard deviations, where appropriate).

3.5.2. Form the feature matrix with the feature vector from each note (columns in the matrix).

4. For the overall best set of features (previously obtained via forward selection), perform dimensionality reduction via PCA:

4.1. In the transformed feature matrix, keep the ones that retain *percVariance*% of the total variance.

5. Implement note clustering:

5.1. Define two clusters (“melodic” and “garbage” clusters), using GMMs.

5.2. Implement cluster optimization with the expectation-maximization algorithm.

5.4. Assign each note to the corresponding cluster.

6. Identify the melodic cluster, based on salience comparison.

(7. Alternatively, perform clustering on the whole note set.)

5.7. Putting It All Together

The melody identification stage is compacted in Algorithm 5.6.

Parameter Name	*Parameter Value*
harmonicCentsRange	40
numHarmonics	6
maxCentsDistMedian	50
percVariance	0.9
numClusters	2
covariance matrices	diagonal
maxIter	1000
minLogDiff	0.0001

Table 5.5. Note clustering parameters.

Algorithm 5.6. Identification of melodic notes.

1. Eliminate ghost harmonically-related notes, according to Algorithm 5.1 (making use of harmonicity and common fate).
2. Select the most salient notes, as in Algorithm 5.2:
   2.1. Delete low-frequency and non-dominant notes.
   2.2. Resolve note overlapping (by removing or truncating notes).
3. Perform melody smoothing, as described in Algorithm 5.3:
   3.1. Implement octave correction.
   3.2. Handle abrupt note transitions based on the definition and analysis of regions of smoothness.
   3.3. Fill in gaps with melody candidates in the allowed range.
   3.4. Restore the timings of previously truncated notes.
4. Eliminate spurious accompaniment notes, as in Algorithm 5.4:
   4.1. Work out abrupt salience transitions.
   4.2. Resolve abrupt duration transitions.

```
5. Execute note clustering, according to Algorithm 5.5[57]:
   5.1. Extract a set of acoustical features.
   5.2. Perform feature selection (recurring to forward selec-
        tion) and dimensionality reduction (using PCA).
   5.3. Cluster the entire set of notes into a melodic and a
        garbage cluster (or five clusters, in case the complete
        note set is used), by training a GMM with the selected
        features.
6. Return the identified melodic notes.
```

5.8. Experimental Results, Analysis and Conclusions

The results of the melody identification stage are presented and discussed in the following paragraphs. Results of both the MIREX' 2004 and 2005 evaluations are presented. The main limitations of the algorithm are discussed, along with hypotheses for further developments.

A. Analysis of Results

Results for the elimination of ghost harmonically-related notes are summarized in Table 5.6. The first four columns (after the ID column) provide information as to the percentage of discarded notes, where the last three concern raw note identification accuracy.

As can be seen in the last line of the third column, 37.8% of the notes present after trajectory segmentation (TS) were deleted during the elimination of ghost notes (EGN). This value is nearly the same in both databases, though the excerpts in the M04 set have a much higher amount of notes because of being longer. In a few excerpts, very high elimination rates were achieved, e.g., Mambo Kings (ID 8) and midi2 (17), with values close to 65%. Other high deletion rates were attained in Claudio Roditi (7) and daisy2 and 3 (IDs 12 and 13). These corresponded to situations where many of the original notes were ghosts. However, very low values appear in samples such as Eliades Ochoa (9) and male opera (19). This seems to be a consequence of excessive curve distance during common modulation analysis, due to several missing values in both the melodic and ghost notes.

[57] This step is not included in practical implementations, since it was not sufficiently robust, as will be seen in the next section. Although those developments are still preliminary, we hope to further work out the encountered difficulties and be able to apply it in a more general framework.

ID	Number of Notes				MRNA		
	TS	EGN	% Elim.	% Kept Melodic	TS	EGN	% Elim. melodic
1	83	47	43.4	34.0	98.0	98.0	0.0
2	85	51	40.0	29.4	82.2	82.2	0.0
3	93	61	34.4	18.0	95.4	95.4	0.0
4	107	75	29.9	21.3	96.6	96.6	0.1
5	100	74	26.0	13.5	85.3	85.3	0.0
6	80	55	31.3	25.5	93.6	93.6	0.0
7	95	40	57.9	47.5	98.8	98.8	0.0
8	71	25	64.8	48.0	93.6	93.6	0.0
9	58	50	13.8	20.0	86.5	86.5	0.0
10	103	78	24.3	14.1	81.1	81.1	0.0
11	107	50	53.3	52.0	95.4	95.4	0.0
12	230	124	46.1	18.5	96.5	96.5	0.0
13	125	67	46.4	16.4	98.1	98.1	0.0
14	241	175	27.4	12.6	81.1	79.5	2.0
15	224	137	38.8	16.1	90.8	90.8	0.0
16	278	150	46.0	26.0	91.7	91.7	0.0
17	142	46	67.6	47.8	98.4	98.4	0.0
18	303	221	27.1	16.7	79.6	78.4	1.5
19	341	288	15.5	21.2	70.2	69.2	0.5
20	392	268	31.6	12.7	72.7	71.5	1.6
21	306	220	28.1	13.2	87.0	87.0	0.0
Avg PDB	89.3	55.1	38.1%	29.4%	91.5%	91.5%	0.0%
Avg M04	258.2	168.0	37.5%	20.1%	86.6%	86.1%	0.6%
Avg	169.7	109.6	37.8%	25.0%	89.2%	88.9%	0.3%

Table 5.6. Results for the elimination of ghost harmonically-related notes.

We took special care so as not to discard true melodic notes. As can be seen in the last column of Table 5.6, an average of only 0.3% of melodic frames where erroneously disposed of in the two databases. Owing to our conservative deletion approach, from all the notes that were kept after ghost elimination, only 25% are melodic. Hence, a high number of non-melodic notes are still present, as illustrated in the fourth column. However, results show that the balance between under and over-elimination was satisfactory.

Although around 38% of the notes are removed, many non-melodic notes remain. Therefore, the notes that convey the main melodic line must be identified among these.

Summary results pertaining to this task are presented in Table 5.7. There, both the *melodic* raw note accuracy (MRNA) and now the *overall* raw note accuracy (ORNA) metrics are evaluated (see Section 2.6.2). The first three columns correspond to the selection of the most salient notes whereas the last four are related to the melody smoothing process.

ID	*Salient Notes*			*Melody Smoothing*			
	MRNA	*MCNA*	*ORNA*	*MRNA*	*MCNA*	*ORNA*	*% Kept melodic*
1	54.1	86.4	53.2	90.5	96.9	89.0	92.3
2	70.6	75.9	56.9	80.5	80.5	65.5	97.9
3	94.2	94.2	91.1	93.6	93.6	90.6	98.1
4	86.9	86.9	67.5	95.6	95.6	74.2	99.0
5	67.4	73.2	51.2	78.1	78.1	57.2	91.5
6	70.4	83.0	61.4	90.8	90.8	80.6	97.1
7	93.5	98.2	87.2	98.2	98.2	91.7	99.5
8	90.3	90.3	83.4	93.6	93.6	86.4	100.0
9	81.3	89.8	64.6	81.3	89.8	64.6	94.1
10	75.1	75.1	53.8	75.1	75.1	53.8	92.6
11	31.6	90.3	31.7	94.2	94.2	92.8	98.7
12	91.3	92.9	79.2	92.0	92.0	81.8	95.3
13	84.7	84.8	84.6	97.5	97.5	97.5	99.4
14	71.6	74.1	66.9	73.6	73.6	70.4	92.6
15	83.1	87.6	61.2	87.3	87.6	64.1	96.2
16	69.6	83.5	67.6	87.4	89.1	85.5	95.4
17	93.3	96.7	91.9	96.7	96.7	95.2	98.2
18	66.5	68.4	59.8	66.6	66.6	64.2	84.9
19	42.5	47.4	40.7	47.4	47.4	44.2	68.1
20	69.7	70.4	61.9	69.6	69.6	65.6	97.4
21	78.9	80.2	69.3	82.3	82.3	74.1	94.6
Avg PDB	*74.1%*	*85.8%*	*63.8%*	*88.3%*	*89.7%*	*76.9%*	*96.4%*
Avg M04	*75.1%*	*78.6%*	*68.3%*	*80.0%*	*80.2%*	*74.2%*	*92.2%*
Avg	*74.6%*	*82.3%*	*66.0%*	*84.4%*	*85.2%*	*75.6%*	*94.4%*

Table 5.7. Results of melody detection: selection of salient notes and melody smoothing.

We can see that good results were achieved in melody smoothing. There, an average accuracy of 84.4 / 75.6% (MRNA / ORNA, respectively) was attained. Also, in several excerpts the system reached almost 100%. Without melody smoothing, the average accu-

racy was 74.6 / 66.0%[58] and so our implementation of the melodic smoothness principle amounts for an average improvement of 9.8 / 9.6%. This is more evident in our test-bed, where the accuracy raised by 14.2 / 13.1% in the two measures.

The relevance of disposing of ghost notes before the identification of melodic notes is confirmed by the clear decrease in performance if melody identification is carried out exactly after note determination (74.2 / 67.3% against 84.4 / 75.6% when ghost note elimination is conducted). In effect, harmonically-related notes abound and some of them are more salient than the real melodic notes. Hence, those are selected, misleading the smoothing algorithm.

Several octave errors were also corrected in the melody smoothing stage, especially in the excerpts from Battlefield Band (ID 11), Pachelbel's Kanon (1) and midi1 (16). In fact, in the conducted experiments the proposed scheme was practically immune to octave errors. In reality, disregarding such errors accuracy after melody smoothing increased to 85.2% (MCNA metric), i.e., an improvement of only 0.8%.

Finally, in the last column, we observe that on an average 94.4% of melodic notes were kept among the ones available after ghost note elimination (see the 6th column of Table 5.6). Only the operatic samples (IDs 18 and 19) stayed much below the average, for the reason that too many non-melodic notes were initially selected, thus misdirecting the melody-smoothing procedure. Indeed, in these cases, long smooth regions with several non-melodic notes are defined, which the smoothing algorithm leaves untouched.

The results obtained for pop/rock and bachata excerpts pleasantly surprised us, since they have strong percussion (Juan Luis Guerra, ID 10), as well as intense guitars (Ricky Martin, 5) with distortion (Avril Lavigne, 6). In effect, heavy percussion is a major cause of pitch detection inaccuracy due to peak interference and masking. In these cases, the allowed trajectory sleeping was most helpful.

Also, the results from the choral sample (2) were very interesting because four simultaneous voices are present, plus orchestral accompaniment. Even so, the algorithm could reasonably well detect the melody, which we defined as corresponding to the soprano. The use of this example contradicts our previous assumptions on the employed excerpts, but we were interested in evaluating a specific situation like this one.

When one single pitch is extracted in each frame, an average accuracy of 62.4 / 63.4% is attained, which improves to 64.1 / 65.0% when octave errors are disregarded. As can be seen, these figures are far behind the ones obtained when multiple pitches are selected. However, concerning the ORNA metric, it shows some improvement over the MRNA measure. Indeed, the net effect of extracting only one pitch per frame is that the number of false negatives increases but, on the other hand, the number of false positives

[58] For fair comparison, we also implemented gap-filling here. The performance without this procedure drops slightly to 71.1 / 64.6%.

decreases. As will be seen, this could be a favorable strategy when the identification of melodic notes is particularly complex, e.g., in samples with low SNR.

Removal of false positives was then addressed. Results are presented in Table 5.8.

ID	*Elim. Spurious*		*Note Clustering*		*Note Clustering (Whole Set)*	
	MRNA	*ORNA*	*MRNA*	*ORNA*	*MRNA*	*ORNA*
1	90.5	89.0	90.5	89.0	90.5	89.0
2	80.5	65.5	82.1	76.9	82.1	72.4
3	93.6	90.6	93.6	90.6	93.6	90.6
4	95.6	74.2	95.6	74.2	95.6	74.2
5	78.1	57.2	76.8	58.6	78.1	57.2
6	90.8	80.6	90.8	87.4	90.8	80.6
7	98.2	91.7	98.2	91.7	98.2	91.7
8	93.6	86.4	93.6	86.4	93.5	86.4
9	81.3	64.6	81.3	71.5	81.3	64.6
10	75.1	53.8	75.1	53.8	75.1	53.8
11	94.2	92.8	94.2	91.2	94.2	92.8
12	95.4	86.8	85.3	78.2	95.4	86.8
13	97.5	97.5	97.5	97.5	97.5	97.5
14	74.1	72.7	73.5	71.3	74.1	72.7
15	83.6	74.7	79.7	85.1	83.6	74.7
16	86.4	85.8	86.4	85.8	89.2	89.4
17	96.7	95.2	96.7	95.2	96.7	95.2
18	66.6	64.2	66.6	64.2	66.6	64.2
19	47.4	45.0	47.4	45.0	47.4	45.0
20	70.1	70.8	69.6	68.8	70.1	70.8
21	83.0	78.1	83.0	78.1	83.0	78.1
Avg PDB	*88.3*	*76.9*	*88.3*	*79.2*	*88.5*	*77.6*
Avg M04	*80.1*	*77.1*	*78.6*	*76.9*	*80.4*	*77.4*
Avg	*84.4*	*77.0*	*83.7*	*78.1*	*84.6*	*77.5*

Table 5.8. Results of the melody detection system: elimination of accompaniment notes.

After the deletion of spurious accompaniment notes, we can see that the ORNA measure improved slightly from 75.6 to 77.0%, i.e., 1.4% increase. This was more apparent in the M04 database, where an improvement of 2.9% was accomplished. A few excerpts were particularly successful, e.g., jazz3 (ID 15), which showed a growth of 10.6%

in the ORNA metric. However, this was achieved at the expense of deleting also a few true melodic notes, which originated a slight decrease in the MRNA measure. This has also happened in other samples, e.g., midi1 (16).

On the other hand, because of note elimination, the original durations of some notes were restored (recall that some of them were truncated when the most salient notes were selected), which led to a slight improvement of MRNA in a few excerpts, e.g., jazz2 (14) or pop1 (20). In the end, the overall MRNA average stayed at 84.4%.

We can see that the overall note accuracy is always lower than the accuracy derived using only the melodic frames. In reality, our method shows a limitation in disposing of false positives (i.e., accompaniment or noisy notes): 31.0% average recall and 52.8% average precision. This is a direct consequence of the fact that the algorithm is biased towards detecting the maximum number of melodic notes, no matter if false positives are included. Moreover, a few extracted notes are slightly longer than the annotated ones. One notorious example is Avril Lavigne's sample (6), where all false positives were eliminated but still the overall accuracy continued below the melodic accuracy[59]. This situation may derive from annotation errors.

As for note clustering, the ORNA measure improved a bit more (1.1%, comparing to the numbers in the elimination of spurious notes, and 2.5%, regarding melody smoothing). This was more evident in our test-bed, where the ORNA metric increased by 2.3% in comparison to the same value after melody smoothing. Namely, Hallelujah (2) showed a striking improvement. In the M04 database, the jazz3 sample (15) also improved substantially. Even so, the average results in the M04 database decayed slightly in both the MRNA and ORNA measures, due to the incorrect removal of true melodic notes. Namely, the accuracy of daisy2 (12) dropped by more than 10%. Other samples were also slightly disturbed by note clustering, e.g., jazz2 (14) and pop1 (20). It is curious that our test-bed was more successful than the M04 database, although longer song excerpts were expected to favor note clustering.

In spite of some improvements, a few excerpts still show several false positives, e.g., Dido (4) and Ricky Martin (5). Indeed, different songs prefer different feature combinations. For example, almost all untrue notes from Juan Luis Guerra's sample were eliminated with a particular feature set. However, the presented best average results were accomplished using the following features (in order of insertion from the forward selection algorithm): harmonic magnitude, relative harmonic magnitude ratio, relative spectral centroid, spectral inharmonicity, spectral centroid, pitch salience, spectral irregularity, harmonic frequency, relative harmonic frequency ratio, spectral skewness, frequency slope in the attack and onset duration.

[59] In fact, even when no false positives are present, the ORNA measure can only equal MRNA values in case no extracted notes span time intervals annotated as silent.

Clustering the whole note set (last two columns) led to similar results: 84.6% / 77.5% accuracy. An interesting outcome of this approach was that, although the ORNA measure stayed practically the same, no true note was incorrectly deleted. In fact, the average MRNA measure increased a bit, contrariwise to the decline observed in the previous clustering mechanism. Therefore, this strategy apparently promotes more stable results. Again, different excerpts prefer different features combinations, but best global results were attained with only these: harmonic magnitude, relative harmonic magnitude ratio, harmonic frequency, relative harmonic frequency ratio and spectral centroid.

As expected, the note-clustering procedure did not prove robust (although clustering in the entire set seemed more secure). In reality, despite its positive impact on several samples, in others the accuracy showed a marked decline. Moreover, the overall best feature set varies from sample to sample and so its identification becomes challenging in a generic context. Thus, for the sake of reliability, the results output by our system are the ones achieved after the elimination of spurious accompaniment notes.

With the purpose of comparing the present results with the ones from MIREX'2004, we also evaluated our method with the exact frequency values used there (i.e., pitch contour metrics, MRPA and ORPA). In this way, the accuracy after eliminating spurious accompaniment notes, taking into consideration only the M04 database, dropped from 80.1 / 77.1% (note metrics - see the first two columns of Table 5.8) to 75.1 / 71.1% (pitch contour metrics), i.e., 5 / 6%.

The defined parameter set was tuned using the excerpts in Table 2.1. Some of the specified thresholds were based on common musical practice (e.g., minimum note duration, maximum pitch interval), as previously defined. However, other values were empirically set, although our initial guesses were usually close to the final values (e.g., the parameters for the elimination of non-dominant notes). As in Section 4.6, and in order to evaluate the influence of parameter variance in the final results, parameter values were individually modified, typically in a [-50%, +50%] range from the defined thresholds (up to 100% in some parameters, e.g., *numTop* parameter, in Table 5.2). In the conducted experiments, we observed a maximum average decrease of 7% in the MRNA metric. Nevertheless, a few individual excerpts had higher variations. For instance, in Ricky Martin's sample (ID 10) we noticed accuracy oscillations of up to +6% and –21%.

The final melody extraction accuracy is obviously affected by the behavior of the first two stages of the system (depicted in Figure 2.1). Particularly, inaccurate pitch detection automatically constrains the maximum achievable performance. However, this has more to do with the nature of the used song excerpts than to algorithmic decisions in pitch detection (for example, strong percussion may cause considerable pitch masking). Regarding the conversion of pitch sequences to musical notes, different parameterizations have an effect on the accuracy of the subsequent stages of the method, particularly the minimum note duration parameter. As discussed in Section 4.6, the maximum decrease in the average melody note accuracy was 6.5%, which resulted from a minimum note

duration of 60 msec.

In order to assess the generality of our approach and the particular parameter tuning, we evaluated it with the test set used in the MIREX'2004 evaluation, which consisted of 10 extra samples. The results achieved for the pitch contour metrics (i.e., MRPA and ORPA) were, respectively, 72.1% and 70.1%. For the note metrics (MRNA and ORNA), the average accuracy was 77.4% and 75.1%, respectively. Hence, the obtained results are only slightly below the ones attained in the M04 training set.

An additional test set was used in the MIREX'2005 evaluation. There, 25 excerpts of 10 to 40 seconds were employed, covering genres such as Rock, R&B, Pop and Jazz [MIREX, 2005; Poliner and Ellis, 2005b]. There, pitch contour accuracy was determined by calculating the percentage of correctly identified frames (where a maximum separation of a quarter-tone from the annotated frequencies was permitted). Also, a granularity of 10 msec was defined (and so we mapped the original 5.8 msec hop size to the required value). In this test-bed the melodic and overall pitch contour accuracy of our method dropped clearly, respectively, to 62.7% and 57.8%. Although we did not have access to the selected excerpts[60], three representative examples were provided by the organizers. These, along with discussions during ISMIR'2005, allow us to deduce that this decrease in efficiency is mostly due to the use of excerpts with lower SNR. In this case, too many non-melodic notes might have been initially selected (a consequence of basing our strategy on the salience principle), which the smoothing algorithm was unable to fix.

The global results for the MIREX'2004 evaluation [Gómez *et al.*, 2006; MIREX, 2004] are summarized in Table 5.9.

Participant	*Overall Raw Pitch Accuracy*			*Overall Chroma Pitch Accuracy*		
	Training Set	*Test Set*	*Training and Test Average*	*Training Set*	*Test Set*	*Training and Test Average*
Paiva	67.2	71.0	69.1	68.0	71.4	69.7
Poliner	66.2	46.1	56.1	66.3	48.0	57.1
Bello	54.3	47.4	50.8	59.3	56.1	57.7
Tappert	42.4	42.0	42.2	55.5	56.3	55.9
Baseline	29.5	36.0	32.7	40.2	44.2	42.2

Table 5.9. Results of the MIREX'2004 evaluation.

[60] The excerpts comprised in the MIREX'2005 database were not made public since the organizers plan to reuse them in future evaluations. This is due to the difficulties in acquiring reliable annotations.

In the previous table, the first column presents the results in the training set (i.e., the ten samples in Table 2.1), the second column corresponds to the test set (ten additional similar excerpts) and the average of the two, which was used for ranking, is presented in the third column. Similar measurements are supplied in the last three columns, except that, there, octave errors are disregarded. Our average results using the two sets are, presently, slightly above (70.6% in the ORPA metric)[61].

The participating algorithms were briefly described in Section 2.4.2. Furthermore, a monophonic pitch tracker proposed in [Cano, 1998], was used to establish a baseline performance for the evaluation. It can be seen that our method surpassed all the others by a healthy amount. Regarding the test set, the difference in accuracy is even more prominent. As to the baseline performance, all systems clearly outperformed it.

In terms of the melodic similarity metric employed in MIREX'2004, our approach scored better than Bello's (the other system that explicitly extracted musical notes) with an average distance of 8.63 over 14.12. As referred to, it is not trivial to correlate those distances to perceptual impressions of similarity. In other words, it is difficult to conclude, using only this metric, whether listeners would recognize the original songs. Hence, this metric is more relevant as a means for comparison of different approaches.

With respect to the MIREX'2005 evaluation [MIREX, 2005], global results are summarized in Table 5.10.

We participated with two algorithms: one corresponding to the present developments, where multiple pitches were extracted in each frame (Paiva MP), and another one where only one pitch was determined (Paiva SP). Unlike all our previous results, the single-pitch approach scored better than the multi-pitch one in the ORPA measure. In reality, our system showed some limitations in this dataset, since most of the used excerpts were pop/rock songs with low SNR. Thus, the scheme adopted for selection of the most salient notes delivered many erroneous notes, which the melody smoothing method was unable to resolve. Given the described difficulties, the single-pitch approach led to more false negatives but fewer false positives, which turned out to be a better strategy.

As can be seen in Table 5.10, Dressler's approach was clearly the best in this evaluation. Indeed, it seems that this method handles reasonably well both the identification of melodic frames (second column) and melody/accompaniment discrimination, besides being the fastest one. On the other hand, ours was the slowest one, mostly because of the employed auditory model, the native Matlab execution and the fact that no optimizations were carried out.

---

[61] In comparison to our system's version by the time of MIREX'2004, some additional implementations were conducted. Namely, the look-ahead and gap-filling procedures were added to the pitch trajectory construction module, frequency-based segmentation was slightly improved (e.g., singer tuning), onset detection was carried out, elimination of spurious notes and note clustering were developed, as well as small improvements to the selection of the most salient notes.

Participant	*ORPA*	*MRPA*	*MCPA*	*Runtime (sec)*
Dressler	71.4	68.1	71.4	32
Ryynänen	64.3	68.6	74.1	10970
Poliner	61.1	67.3	73.4	5471
Paiva (SP)	61.1	58.5	62.0	45618
Marolt	59.5	60.1	67.1	12461
Paiva (MP)	57.8	62.7	66.7	44312
Goto	49.9	65.8	71.8	211
Vincent1	47.9	59.8	67.6	?
Vincent2	46.4	59.6	71.1	251
Brossier	3.2	3.9	8.1	41

Table 5.10. Results of the MIREX'2005 evaluation.

It can also be observed that the actual Goto's system behaved much better than Tappert and Batke's implementation used in MIREX'2004. Although we only evaluated the probabilistic front-end for pitch detection, we can infer that our pitch detection results (presented in Section 3.6) are also below the ones obtained in the actual system. As previously referred to, both Tappert and Batke's version and ours seem to have missed some sort of implementation peculiarities.

In the same table, the results for the last four participants are not directly comparable since, in these, melodic discrimination is not conducted, i.e., an F0 value is output in each frame. Furthermore, scores for Brossier are artificially low due to an unresolved algorithmic issue.

B. Limitations of the Algorithm and Possible Improvements

Our approach seems relatively style-independent, since the pitch accuracy did not differ significantly among the different excerpts (excerpt for opera). However, the algorithm had more difficulties in songs with low signal-to-noise ratio, as confirmed by its lower performance in the MIREX'2005 evaluation. Also, excerpts with strong vibrato, like the opera samples, put additional obstacles on the pitch detection and note determination stage, which was reflected on the performance of melody identification.

The main drawback of the melody identification stage is in the discrimination between the melody and the accompaniment. Indeed, our attempts towards note clustering

lacked robustness, as the best set of features varies from sample. Moreover, some particular feature combinations simply cannot discriminate between true notes and false positives, causing a notorious drop in melody detection accuracy. Therefore, for the time being, robustness cannot be assured after the elimination of spurious notes. In any case, longer song excerpts could possibly improve the behavior of note clustering.

Feature extraction in a polyphonic context is also a challenging issue. In effect, some harmonic magnitudes may be unreliable due to spectral collisions. Hence, corrupted components should be discarded and clustering should be attempted following a missing feature strategy (e.g., [Eggink and Brown, 2003]).

Also, note clustering via timbral features places some difficulties on melody extraction in songs where the solo moves from instrument to instrument, e.g., jazz pieces in which different instruments alternate the lead. In such cases, when the soloist changes, the notes from the dominant instrument will be erroneously discarded. In fact, timbre is not the only meaningful feature for melody grouping; as previously referred to, the highness of each individual part, as well as proximity and intensity, play an important role. Anyway, in the available test-beds the lead instrument is fixed, which theoretically allows for melody extraction recurring to note clustering. Also, we should point out that the previous algorithms (selection of salient notes and melody smoothing) are transparent to eventual soloist changes, having, however, the problem of delivering false positive notes.

Regarding execution time, our approach clearly shows a weak point here. This is a consequence of the use of an expensive pitch detection scheme. In reality, about 97% of the total execution time is spent in the first stage of the algorithm, with particular incidence on the derivation of the cochleagram and correlogram in each frame. Hence, our method was considerably slower than the fastest one at both MIREX'2004 and MIREX'2005. The presented figures, though not directly comparable because of differences in operating systems and languages (e.g., Windows vs Linux, Matlab vs C), show substantial discrepancies. Anyway, computational time is not yet a major issue, since this field of research is still struggling for accurate, general and robust results. However, the tremendous time inefficiency of the auditory front-end raises the question of its future feasibility for large music collections.

C. Other Possible Improvements

As referred to, the melody smoothing procedure is unable to fix situations were too many erroneous notes are selected. This should be worked out in future developments, e.g., by finding the best melodic path through the set of available notes, much in the same way as Ryynänen and Klapuri do [Ryynänen and Klapuri, 2005b]. In addition, higher-level information could be further exploited in our system, namely with recourse to key and tonality estimation, probabilities of note transitions, or application of voice-leading rules.

Chapter 6

Conclusions and Perspectives

"Now this is not the end. It is not even the beginning of the end. But it is, perhaps, the end of the beginning."

Winston Churchill, "Speech after the British defeat of the German Afrika Korps in Egypt", November 10, 1942

Having reached the end of this dissertation, our first thought is that this is, at best, the end of an initial stage towards melody detection in polyphonic audio. In fact, any task dealing with content analysis of polyphonic musical signals, and melody extraction in particular, is inherently problematical. Tasks such as pitch detection in complex mixtures with strong percussive sounds, accurate conversion of pitch sequences into musical notes or segregation of melodic notes entail intricate research issues, for which we have offered a humble contribution. The attained results, though motivating, show that there is room for improvement.

In this chapter, we summarize the work carried out throughout this document and the main conclusions that can be drawn from it. A not so specific analysis is conducted, since several aspects were already discussed during the evaluation sections in each of the previous chapters. Therefore, only a few key technical matters are addressed here.

Based on the main encountered difficulties and on the current trends on MIR research, our general perspectives for future work are presented, in terms of possible improvements to the current approach and its extension towards music retrieval.

6.1. Summary and Conclusions

In this dissertation, we presented a system for melody detection in polyphonic musical signals. Besides other possible applications, this is a main concern for MIR tools such as QBH in "real-world" music databases. Existing work in this field is presently almost en-

tirely confined to the MIDI domain, and so we believe to have given an interesting contribution to the area, with encouraging results.

Our system starts with a melody-oriented pitch detection methodology, where an auditory-model-based pitch detector is adopted and extended for multiple-pitch extraction. One of our basis assumptions is that melodic notes are usually salient in polyphonic mixtures. Hence, selecting a few of the most intense F0s in each frame leads to satisfactory results. However, in songs with low SNR, peak masking occurs more prominently, which is mostly due to percussive sounds. Experiments were performed towards frame-wise percussion elimination but the accomplished results were not convincing. In fact, this is a complex subject that needs further attention in the future. Anyway, this shortcoming was partly attenuated during pitch track construction, where track inactivity is allowed, making it possible to restore undetected F0s.

Unlike most other melody extraction schemes, our method explicitly identifies musical notes with precise pitches and timings, something that is not attended to in most related research. The achieved results, despite showing that there are opportunities for improvement, are positive. The main drawbacks of the algorithm result from its reliance on the definition of a minimum note duration and from the limitations of onset detectors in polyphonic contexts. The former gave rise to difficulties on the segmentation of pitch tracks with extreme vibrato, such as in opera pieces. The latter placed obstacles on the accurate segmentation of consecutive notes at the same pitch.

As a result of our multi-pitch detection strategy, several notes are created, among which the main melodic line must be identified. This is not a trivial task since many aspects of auditory organization influence the perception of melody by humans, for instance in terms of the pitch, timbre and intensity content in a given musical mixture. In this way, we resorted mostly to aspects of intensity, where the most salient notes at each time are first selected, and frequency proximity, where the initial melodic contour is smoothed out. The obtained results were quite satisfactory in the used test-bed. Nevertheless, in sound signals where many salient non-melodic notes are present, e.g., musical pieces with low SNR, the smoothing procedure experienced difficulties in replacing the incorrect notes with the melodic ones. In reality, long smooth regions are validated, regardless of containing a high number of erroneous notes or not. This seems to have been the case in the MIREX'2005 evaluation.

Additionally, we tackled the problem of false positives. As expected, this proved to be difficult and so only slight improvements were achieved. Spurious accompaniment notes that appear for brief moments during pauses between melodic notes were reasonably well dealt with. However, note clustering, aiming to delete accompaniment notes that are output when the solo stops, lacked robustness. Indeed, the best feature set varied from excerpt to excerpt. Also, a more effective clustering process would possibly require a higher number of notes in each song sample, which is not the case.

To sum up, we most likely need many years of intensive research before sufficiently robust, accurate and efficient melody detection algorithms become available for commercial purposes, much in the same way that speech recognition systems only attained a minimum acceptable performance after several years.

6.2. Perspectives for Future Research

With respect to future work, we plan to further work out some of the described limitations, namely in what concerns the generality of the melody identification module. As referred to, the algorithm shows some difficulties when the assumption that the melodic notes are usually salient in the mixture fails. To this end, higher-level cognitive information, e.g., memories and expectations or prior-knowledge relating to the properties of musical events, could be exploited, mimicking in this manner the human music-listening experience to some extent.

In this way, we could take advantage of pattern detection and matching in music. In fact, the music-listening mechanism is actively accompanied by memorization and recognition of patterns, which create expectations on what is to come. Therefore, the recognized musical patterns have predictive power, which could be valuable for melody identification, namely for post-processing tasks such as error detection and correction. In our algorithm, the selection of the notes carrying the melody could be supported by the detected patterns. For example, if a sequence of notes is very close to a previously detected succession, the next note could be selected as the one that best continues the pattern in cause (in an exact or fuzzy way, dealing with different notes that might appear in the succession). This in turn requires similarity metrics such as the edit distance (Section 2.6.2). Likewise, statistics regarding the most common sequences of notes in a given piece could be used.

Context information could also be added as an improvement to the system. For example, the tonality of the piece under analysis, added to musicological information regarding the preferred notes for particular keys, as well as the use of note transition probabilities, could be beneficial for resolving ambiguities in note selection. Moreover, statistics pertaining to the most common sequences of notes for pieces played in particular keys could be utilized.

Additionally, meter and rhythm information could be exploited to support melody identification. Indeed, notes starting in synchronism with strong beats are more likely to be correct.

The meaningful integration of such diverse knowledge sources is usually not trivial. Hence, probabilistic methods or blackboard systems could be employed as decision-aid modules.

The suggested developments are likely to improve melody/accompaniment discrimination. Moreover, this could also be accomplished by conducting note clustering with longer song excerpts, as previously referred to.

The last point stresses the urgent need of larger, longer and more varied test-beds. In reality, the available standard compilations (from MIREX'2004 and 2005) are very short in number of song excerpts and also lack variety, in spite of the efforts to make them sufficiently significant. This is a result of the difficulties in acquiring reliably annotated songs. In this way, multi-track recordings seem like a good alternative. We are presently establishing contacts towards this end.

Also, as previously mentioned, we have devised a general-purpose mechanism for melody detection. However, every musical style has its own peculiarities. Thus, in the current state of events, research could also evolve by focusing on approaches targeting specific musical set-ups. In effect, "methods can be different according to the complexity of music (monophonic or polyphonic), the genre (classical with melodic ornamentations, jazz with singing voice, etc.) or the representation of music (audio, midi, etc.)" [Gómez *et al.,* 2006]. This is reflected in the fact that the performances of different methods depend on distinct musical characteristics, e.g., some perform better in singing excerpts whereas others prefer instrumental solos, some do a good job in music with extreme dynamics while others fail there, others are more robust in excerpts with low SNR and still others are more successful in songs of specific genres. Similarly, music outside the common-practice Western canon should be attended to.

An obvious follow-up to our work would be the construction of a prototype for query-by-melody in an audio database. Indeed, it would be interesting to evaluate the robustness of QBM systems to automatically created and imperfect melody databases, such as the ones obtained by existing melody extraction algorithms. It is often argued that realistic retrieval by similarity is only possible in the query-by-example domain, and so an application gap would be filled if sufficient accuracy were attained.

The subject of imperfectly extracted melodies is key for robust QBM. Actually, current systems operate almost exclusively on clearly defined melodies, available in a separate MIDI channel. For audio QBM, robustness to inaccurate melodies, with inexact timings, missing and extra notes or semi-tone errors, is of primary concern. These topics have been addressed to some extent recently (e.g., [Pikrakis and Theodoridis, 2005; Song *et al.,* 2002]), but the results confirm the need of more accurate and general systems.

Besides the technical issues relating to query transcription and matching, other higher-level questions have to be unequivocally answered, namely regarding the rigorous definition of what a query and a answer are and what constitutes similarity [Uitdenbogerd *et al.,* 2000]. Answers to these questions depend on users' needs, e.g., whether the query should be hummed, sung, whistled or played in an instrument; monophonic or polyphonic; whether robustness to off-key queries or queries with miss-

ing or extra notes should be a requirement; or whether the list of responses is intended for plagiarism detection or spotting of half-remembered songs.

Additionally, the automatic summarization of songs is fundamental for both query matching and presentation of results. Research under this topic is already evolving, with promising results (e.g., [Peeters *et al.*, 2002]).

BIBLIOGRAPHY

Agostini G., Longari M. and Pollastri E. (2001). "Musical Instrument Timbres Classification with Spectral Features", *In Proc. IEEE Workshop on Multimedia Signal Processing – MMSP'2001*.

Alghoniemy M. and Tewfik A. (2000). "Personalized Music Distribution", *In Proc. IEEE International Conference on Acoustics, Speech and Signal Processing – ICASSP'2000*.

ANSI (1973). *Psychoacoustical Terminology*, American National Standards Institute (ANSI), New York, USA.

ANSI (1994). *ANSI S1.1-1994*, American National Standard Acoustical Terminology, Acoustical Society of America.

Askenfelt A. (1979). "Automatic Notation of Played Music: the VISA Project", *Fontes Artis Musicae*, Vol. XXVI, No. 2, pp. 109-118.

Aucouturier J.-J. and Pachet F. (2004). "Tools and Architectures for the Evaluation of Similarity Measures: Case Study of Timbre Similarity", *In Proc. International Conference on Music Information Retrieval – ISMIR'2004*.

Bainbridge *et al.*, 1999Bainbridge D., Nevill-Manning C. G., Witten I. H., Smith L. A. and McNab R. J. (1999). "Towards a Digital Library of Popular Music", *In Proc. ACM International Conference on Digital Libraries – DL'99*, pp. 161-169.

Barlow H. and Morganstern S. (1948). *A Dictionary of Musical Themes*, New York, Crown.

Batke J.-M., Eisenberg G., Weishaupt, P. and Sikora T. (2004). "A Query by Humming System Using MPEG-7 Descriptors", *In Proc. 116th Audio Engineering Society Convention – AES116*.

Bello J. P. (2003). *Towards the Automatic Analysis of Simple Polyphonic Music: A Knowledge-based Approach*, PhD Thesis, Department of Electronic Engineering, Queen Mary, University of London, UK.

Bello J. P., Daudet L., Abdallah S., Duxbury C., Davies M. and Sandler M. (2005). "A Tutorial on Onset Detection in Music Signals", *IEEE Transactions on Speech and Audio Processing*, Vol. 13, No. 5, pp. 1035–1047.

Berenzweig A., Logan B., Ellis D. and Whitman B. (2003). "A Large-Scale Evaluation of Acoustic and Subjective Music Similarity Measures", *In Proc. International Conference on Music Information Retrieval – ISMIR'2003*.

Birmingham W. P., Dannenberg R. B., Wakefield G. H., Bartsch M., Bykowski D., Maz-

zoni D., Meek C., Mellody M. and Rand W. (2001). "MusArt: Music Retrieval Via Aural Queries", *In Proc. International Symposium on Music Information Retrieval – ISMIR'2001.*

Bishop C. M. (1995). *Neural Networks for Pattern Recognition*, Oxford University Press.

Brandão A. (2004). *Teoria Musical - Melodia*, URL: http://www.allegrobr.com/biblioteca/artigo.php?id=16, available by July 31, 2006 (in Portuguese).

Bregman A. S. (1990). *Auditory Scene Analysis: The Perceptual Organization of Sound*, MIT Press.

Brossier P., Bello J. P. and Plumbey M. D. (2004). "Fast Labelling of Notes in Music Signals", *In Proc. International Conference on Music Information Retrieval – ISMIR'2004.*

Byrd D. (2002). "The History of ISMIR - A Short Happy Tale", *D-Lib Magazine*, Vol. 8, No. 1, URL: http://www.dlib.org/dlib/november02/11inbrief.html#BYRD.

Cahill M. and Ó Maidín D. (2005). "Melodic Similarity Algorithms - Using Similarity Ratings for Development and Early Evaluation", *In Proc. International Conference on Music Information Retrieval – ISMIR'2005.*

Cano P. (1998). "Fundamental Frequency Estimation in the SMS Analysis", *In Proc. COST G6 Conference on Digital Audio Effects – DAFx'98.*

Casey M. A. and Westner A. (2000). "Separation of Mixed Audio Sources by Independent Subspace Analysis", *In Proc. International Computer Music Conference – ICMC'2000.*

Celma O., Ramírez M. and Herrera P. (2005). "Foafing the Music: A Music Recommendation System Based on RSS Feeds and User Preferences", *In Proc. International Conference on Music Information Retrieval – ISMIR'2005.*

Chafe C., Mont-Reynaud B. and Rush L. (1982). "Toward an Intelligent Editor of Digital Audio: Recognition of Musical Constructs", *Computer Music Journal*, Vol. 6, No. 1, pp. 30-41.

Chafe C., Jaffe D., Kashima K., Mont-Reynaud B. and Smith J. (1985). "Techniques for Note Identification in Polyphonic Music", *In Proc. International Computer Music Conference – ICMC'85*, pp. 399-406.

Chafe C. and Jaffe D. (1986). "Source Separation and Note Identification in Polyphonic Music", *In Proc. International Conference on Acoustics, Speech, and Signal Processing – ICASSP'86*, pp. 1289-1292.

Chai W. (2001). *Melody Retrieval on the Web*, MSc Thesis, School of Architecture and Planning, Massachusetts Institute of Technology, USA.

Clarisse L. P., Martens J. P., Lesaffre M., De Baets B., De Meyer H. and Leman M. (2002). "An Auditory Model Based Transcriber of Singing Sequences", *In Proc. International Conference on Music Information Retrieval – ISMIR'2002.*

de Cheveigné A. and Kawahara H. (1999). "Multiple Period Estimation and Pitch Perception Model", *Speech Communication*, Vol. 27, pp. 175–185.

de Cheveigné A. and Kawahara H. (2002). "YIN, a Fundamental Frequency Estimator for Speech and Music", *Journal of the Acoustic Society of America*, Vol. 111, No. 4, pp. 1917-1930.

de la Cuadra P., Master A. and Sapp C. (2001). "Efficient Pitch Detection Techniques for Interactive Music", *In Proc. International Computer Music Conference – ICMC'2001.*

Doraisamy S. and Rüger S. M. (2001). "An Approach Towards a Polyphonic Music Retrieval System", *In Proc. International Symposium on Music Information Retrieval – ISMIR'2001.*

Dorritie F. (2000). *Essentials of Music for Audio Professionals*, Artistpro.

Doval B. and Rodet X. (1991). "Estimation of Fundamental Frequency of Musical Sound Signals", *In Proc. IEEE International Conference on Acoustics, Speech and Signal Processing – ICASSP'91*, pp. 3657–3660.

Dovey M. (2001). "A Technique for "Regular Expression" Style Searching in Polyphonic Music", *In Proc. International Symposium on Music Information Retrieval – ISMIR'2001.*

Dowling W. J. (1978). "Scale and Contour: Two Components of a Theory of Memory for Melodies", *Psychological Review*, Vol. 85, No. 4, pp. 341-354.

Downie J. S. (2002). "Report on ISMIR 2002 Conference Panel I: Music Information Retrieval Evaluation Frameworks", *D-Lib Magazine*, Vol. 8, No. 1, URL: http://www.dlib.org/dlib/november02/11inbrief.html#DOWNIE.

Dressler K. (2005). "Extraction of the Melody Pitch Contour from Polyphonic Audio", *In Proc. Music Information Retrieval Exchange – MIREX'2005.*

Dunn J. W. (2000). "Beyond VARIATIONS: Creating a Digital Music Library", *In Proc. International Symposium on Music Information Retrieval – ISMIR'2000.*

Effelsberg W. (1998). "Music in Multimedia Systems", *IEEE Multimedia*, Vol. 5, No. 3, pp. 16, Guest Editor's Introduction.

Eggink J. and Brown G. J. (2003). "Application of Missing Feature Theory to the Recognition of Musical Instruments in Polyphonic Audio", *In Proc. International Conference on Music Information Retrieval – ISMIR'2003.*

Eggink J. and Brown G. J. (2004). "Extracting Melody Lines from Complex Audio", *In Proc. International Conference on Music Information Retrieval – ISMIR'2004.*

Ellis D. (1992). *A Perceptual Representation of Sound*, MSc Thesis, Department of Electrical Engineering and Computer Science, Massachusetts Institute of Technology, USA.

Ellis D. (1996). *Prediction-Driven Computational Auditory Scene Analysis*, PhD Thesis, Department of Electrical Engineering and Computer Science, Massachusetts Institute of Technology, USA.

Eronen A. (2001). *Automatic Musical Instrument Recognition*, MSc Thesis, Tampere University of Technology, Finland.

Fingerhut M. (1999). "The IRCAM Multimedia Library: A Digital Music Library", *In Proc. IEEE Forum on Research and Technology Advances in Digital Libraries – IEEE ADL'99*, pp. 129-140.

Flanagan J. L. and Golden R. M. (1966). "Phase vocoder", *The Bell System Technical Journal*, Vol. 45, pp. 1493-1509.

Francès R. (1958). La Perception de la Musique (*The Perception of Music*), Dowling W. J. translation, 1988, Erlbaum, Hillsdale, New Jersey.

Francu C. and Nevill-Manning C. G. (2000). "Distance Metrics and Indexing Strategies for a Digital Library of Popular Music", *In Proc. IEEE International Conference on Multimedia and Expo – ICME'2000*, pp. 889-892.

Fujinaga I. (1998). "Machine Recognition of Timbre Using Steady-State Tone of Acoustic Musical Instruments", *In Proc. International Computer Music Conference – ICMC'98*.

Futrelle J. and Downie J. S. (2003). "Interdisciplinary Research Issues in Music Information Retrieval: ISMIR 2000-2002", *Journal of New Music Research*, Vol. 32, No. 2, pp. 121-131.

Gerhard D. (1998). *Computer Music Analysis*, Technical Report, School of Computing Science, Simon Fraser University, Canada.

Gerhard D. (2000). *Audio Signal Classification*, PhD Depth Paper, School of Computing Science, Simon Fraser University, Canada.

Gerhard D. (2003). *Pitch Extraction and Fundamental Frequency: History and Current Techniques*, Technical Report, Department of Computer Science, University of Regina, Canada.

Ghias A., Logan J., Chamberlin D. and Smith B. C. (1995). "Query by Humming: Musical Information Retrieval in an Audio Database", *In Proc. ACM Multimedia Conference – Multimedia'95*.

Gold B. and Rabiner L. (1969). "Parallel Processing Techniques for Estimating Pitch Periods of Speech in the Time Domain", *Journal of the Acoustical Society of America*, Vol. 46, No. 2, pp. 442-448.

Gomes A. R. (2005). "Festivais de Verão: Para Além da Música, Uma Experiência", *Revista XIS – Jornal Público*, August 6, 2005, pp. 18-19 (in Portuguese).

Gómez E. (2002). *Melodic Description of Audio Signals for Music Content Processing*, Doctoral Pre-Thesis, Department of Technology, Pompeu Fabra University, Barcelona, Spain.

Gómez E., Klapuri A. and Meudic B. (2003). "Melody Description and Extraction in the Context of Music Content Processing", *Journal of New Music Research*, Vol. 32, No.

1, pp. 23-40.

Gómez E., Streich S., Ong B., Paiva R. P., Tappert S., Batke J.-M., Poliner G., Ellis D. and Bello J. P. (2006). *A Quantitative Comparison of Different Approaches for Melody Extraction from Polyphonic Audio Recordings*, Technical Report, Music Technology Group, Pompeu Fabra University, Spain.

Goto M. (2000). "A Robust Predominant-F0 Estimation Method for Real-Time Detection of Melody and Bass Lines in CD Recordings", *In Proc. IEEE International Conference on Acoustics, Speech and Signal Processing – ICASSP'2000*.

Goto M. (2001). "A Predominant-F0 Estimation Method for CD Recordings: MAP Estimation Using EM Algorithm for Adaptive Tone Models", *In Proc. IEEE International Conference on Acoustics, Speech and Signal Processing – ICASSP'2001*.

Grachten M., Arcos J. L. and López de Mántaras R. (2002). "A Comparison of Different Approaches to Melodic Similarity", *In Proc. International Conference on Music and Artificial Intelligence – ICMAI'2002*.

Hainsworth S. W. (2001). *Analysis of Musical Audio for Polyphonic Transcription*, 1st Year Report, Department of Engineering, University of Cambridge.

Handel S. (1989). *Listening - An Introduction to the Perception of Auditory Events*, MIT Press.

Hartmann W. M. (1997). *Signals, Sound and Sensation*, AIP Press.

Haus G. and Pollastri E. (2001). "An Audio Front End for Query-by-Humming Systems", *In Proc. International Symposium on Music Information Retrieval – ISMIR'2001*.

Hawley M. (1993). *Structure Out of Sound*, PhD Thesis, Media Laboratory, Massachusetts Institute of Technology, USA.

Hermansky H., Morgan N. and Hirsch H. G. (1993). "Recognition of Speech in Additive and Convolutional Noise Based on RASTA Spectral Processing", *In Proc. IEEE International Conference on Acoustics, Speech and Signal Processing – ICASSP'93*, pp. 83-86.

Hess W. (1983). *Pitch Determination of Speech Signals: Algorithms and Devices*, Springer-Verlag.

Hofmann-Engl L. (2003). *Melodic Similarity and Transformations: A Theoretical and Empirical Approach*, PhD Thesis, Department of Psychology, Keele University, UK.

Huron D. (1997). "Humdrum and Kern: Selective Feature Encoding", In Selfridge-Field E. (ed.): *Beyond MIDI: The Handbook of Musical Codes*, pp. 275-401, MIT Press.

Huron D. (2000). "Perceptual and Cognitive Applications in Music Information Retrieval", *In Proc. International Symposium on Music Information Retrieval – ISMIR'2000*.

Huron D. (2001). "Tone and Voice: A Derivation of the Rules of Voice-Leading from Perceptual Principles", *Music Perception*, Vol. 19, No. 1, pp. 1-64.

Kageyama T., Mochizuki K. and Takashima Y. (1993). "Melody Retrieval with Humming", *International Computer Music Conference – ICMC'93*.

Kashino K., Nakadai K., Kinoshita T. and Tanaka H. (1995). "Organization of Hierarchical Perceptual Sounds: Music Scene Analysis with Autonomous Processing Modules and a Quantitative Information Integration Mechanism", *In Proc. International Joint Conference on Artificial Intelligence – IJCAI'95*, pp. 158-164.

Kassler M. (1966). "Toward Musical Information Retrieval", *Perspectives of New Music*, Vol. 4, No. 2, pp. 59-67.

Katayose H. and Inokuchi S. (1989). "The Kansei Music System", *Computer Music Journal*, Vol. 13, No. 4, pp. 72-77.

Kilian J. and Hoos H. H. (2002). "Voice Separation - A Local Optimisation Approach", *In Proc. International Conference on Music Information Retrieval – ISMIR'2002*.

Kim Y. E., Chai W., Garcia R. and Vercoe B. (2000). "Analysis of a Contour-Based Representation for Melody", *In Proc. International Symposium on Music Information Retrieval – ISMIR'2000*.

Kitahara T., Goto M., Komatani K., Ogata T. and Okuno H. G. (2005). "Instrument Identification in Polyphonic Music: Feature Weighting with Mixed Sounds, Pitch-Dependent Timbre Modeling, and Use of Musical Context", *In Proc. International Conference on Music Information Retrieval – ISMIR'2005*.

Klapuri A. P. (1999). "Sound Onset Detection by Applying Psychoacoustic Knowledge", *In Proc. IEEE International Conference on Acoustics, Speech and Signal Processing – ICASSP'99*.

Klapuri A. P. and Astola J. T. (2002). "Efficient Calculation of a Physiologically-Motivated Representation for Sound," *In Proc. IEEE International Conference on Digital Signal Processing – DSP'2002*.

Klapuri A. P. (2003). "Multiple Fundamental Frequency Estimation Based on Harmonicity and Spectral Smoothness", *IEEE Transactions on Speech and Audio Processing*, Vol. 11, No. 6, pp. 804–816.

Klapuri A. P. (2004). *Signal Processing Methods for the Automatic Transcription of Music*, PhD Thesis, Tampere University of Technology, Finland.

Klapuri A. P. (2005). "A Perceptually Motivated Multiple-F0 Estimation Method", *In Proc. IEEE Workshop on Applications of Signal Processing to Audio and Acoustics – WASPAA'2005*.

Kornstadt A. (1998). "Themefinder: A Web-based Melodic Search Tool", In Hewlett W. and Selfridge-Field E. (eds.): *Melodic Similarity Concepts, Procedures and Applications*, MIT Press.

Kunieda N., Shimamura T. and Suzuki J. (1996). "Robust Method of Measurement of

Fundamental Frequency by ACLOS - Autocorrelation of Log Spectrum", *In Proc. IEEE International Conference on Acoustics, Speech and Signal Processing – ICASSP'96*.

Lahat A., Niederjohn R. J. and Krubsack D. A. (1987). "A Spectral Autocorrelation Method for Measurement of the Fundamental Frequency of Noise-Corrupted Speech", *IEEE Transactions on Acoustics, Speech and Signal Processing*, Vol. 35, No. 6, pp. 741-750.

Lemström K. and Perttu S. (2000). "SEMEX - An Efficient Music Retrieval Prototype", *In Proc. International Symposium on Music Information Retrieval – ISMIR'2000*.

Lesaffre M., Leman M., De Baets B. and Martens J.-P. (2004). "Methodological Considerations Concerning Manual Annotation of Musical Audio in Function of Algorithm Development", *In Proc. International Conference on Music Information Retrieval – ISMIR'2004*.

Levitin D. J. (1999). "Memory for Musical Attributes", In Cook P. R (ed.): *Music, Cognition and Computerized Sound*, pp. 214-215, MIT Press.

Licklider J. C. R. (1951). "A Duplex Theory of Pitch Perception", *Experentia*, Vol. 7, pp. 128-133.

Logan B. and Salomon A. (2001). "A Music Similarity Function Based on Signal Analysis", *In Proc. IEEE International Conference on Multimedia and Expo – ICME'2001*.

Lyon R. F. (1982). "A Computational Model of Filtering, Detection and Compression in the Cochlea", *In Proc. IEEE International Conference on Acoustics, Speech and Signal Processing – ICASSP'82*, pp. 1282-1285.

Maher R. C. (1989). *An Approach for the Separation of Voices in Composite Musical Signals*, PhD Thesis, College of Engineering, University of Illinois, Urbana-Champaign.

Maher R. C. (1990). "Evaluation of a Method for Separating Digitized Duet Signals", *Journal of the Audio Engineering Society*, Vol. 38, No. 12, pp. 956-979.

Maher R. C. and Beauchamp J. W. (1993). "Fundamental Frequency Estimation of Musical Signals Using a Two-Way Mismatch Procedure", *Journal of the Acoustical Society of America*, Vol. 95, No. 4, pp. 2254-2263.

Marolt M. (2004). "On Finding Melodic Lines in Audio Recordings", *In Proc. International Conference on Digital Audio Effects – DAFx'04*.

Marolt M. (2005). "Audio Melody Extraction Based on Timbral Similarity of Melodic Fragments", *In Proc. International Conference on "Computer as a Tool" – EUROCON'2005*.

Martin K. D. (1996). "Automatic Transcription of Simple Polyphonic Music: Robust Front End Processing", *In Proc. 3rd Joint Meeting of the Acoustical Societies of America and Japan*.

Martin K. D. (1999). *Sound-Source Recognition: A Theory and Computational Model*, PhD

Thesis, Department of Electrical Engineering and Computer Science, Massachusetts Institute of Technology, USA.

Martins L. G. (2001). *PCM to MIDI Transposition*, MSc Thesis, Department of Electrical and Computer Engineering, University of Porto, Portugal.

McNab R. J., Smith L. A. and Witten I. H. (1996a). "Signal Processing for Melody Transcription", *In Proc. Australasian Computer Science Conference – ACSC'96*.

McNab R. J., Smith L. A., Witten I. H., Henderson C. L. and Cunningham S. J. (1996b). "Towards the Digital Music Library: Tune Retrieval from Acoustic Input", *In Proc. ACM International Conference on Digital Libraries – DL'96*.

Medan J., Yair E. and Chazan D. (1991). "Super Resolution Pitch Determination of Speech Signals", *IEEE Transactions on Signal Processing*, Vol. 39, No. 1, pp. 40-48.

Meddis R. and Hewitt M. J. (1991). "Virtual Pitch and Phase Sensitivity of a Computer Model of the Auditory Periphery: I. Pitch Identification", *Journal of the Acoustical Society of America*, Vol. 89, No. 6, pp. 2866-2882.

Meddis R. and O'Mard L. (1997). "A Unitary Model of Pitch Perception," *Journal of the Acoustical Society of America*, Vol. 102, No. 3, pp. 1811-1820.

Meek C. and Birmingham W. P. (2001). "Thematic Extractor", *In Proc. International Symposium on Music Information Retrieval – ISMIR'2001*.

Meyer L. (1956). *Emotion and Meaning in Music*, The University of Chicago Press.

Minami K., Akutsu A., Hamada H. and Tomomura Y. (1998). "Video Handling with Music and Speech Detection", *IEEE Multimedia*, Vol. 5, No. 3, pp. 17-25.

MIREX (2004). "ISMIR'2004 Audio Description Contest (or 1st Music Information Retrieval Evaluation Exchange – MIREX'2004)", *International Conference on Music Information Retrieval – ISMIR'2004*, URL: http://ismir2004.ismir.net/ISMIR_Contest.html, available by July 31, 2006.

MIREX (2005). *The 2nd Music Information Retrieval Evaluation Exchange – MIREX'2005*, URL: http://www.music-ir.org/mirex2005/, available by July 31, 2006.

Moorer J. A. (1977). "On the Transcription of Musical Sound by Computer", *Computer Music Journal*, Vol. 1, No. 4, pp. 32-38.

MPEG-7 (2004). *MPEG-7 Overview (version 10)*, URL: http://www.chiariglione.org/mpeg/standards/mpeg-7/mpeg-7.htm, available by July 31, 2006.

Nabney I. and Bishop C. (1996). *Netlab Neural Network Software*, URL: http://www.ncrg.aston.ac.uk/netlab/index.php, available by July 31, 2006.

Nettheim N. (1992). "On the Spectral Analysis of Melody", *Journal of New Music Research*, Vol. 21, pp. 135-148.

Noll A. M. (1967). "Cepstrum Pitch Determination", *Journal of the Acoustical Society of*

America, Vol. 41, No. 2, pp. 293-309.

Orpen K. S. and Huron D. (1992). "Measurement of Similarity in Music: A Quantitative Approach for Non-Parametric Representations", *Computers in Music Research*, Vol. 4, pp. 1-44.

Pachet F. and Cazaly D. (2000). "A Taxonomy of Musical Genres", *In Proc. International Conference on Recherche d'Information Assistée par Ordinateur – RIAO'2000*.

Pampalk E. (2001). *Islands of Music: Analysis, Organization and Visualization of Music Archives*, MSc Thesis, Department of Software Technology and Interactive Systems, Vienna University of Technology, Austria.

Parker C. (2005). "Applications of Binary Classification and Adaptive Boosting to the Query-by-Humming Problem", *In Proc. International Conference on Music Information Retrieval – ISMIR'2005*.

Pauws S. and Wijdeven S. (2005). "User Evaluation of a New Interactive Playlist Generation Concept", *In Proc. International Conference on Music Information Retrieval – ISMIR'2005*.

Peeters G., La Burthe A. and Rodet X. (2002). "Toward Automatic Music Audio Summary Generation from Signal Analysis", *In Proc. International Conference on Music Information Retrieval – ISMIR'2002*.

Perdigão F. (1997). *Modelos do Sistema Auditivo Periférico no Reconhecimento Automático da Fala*, Department of Electrical Engineering, University of Coimbra, Portugal (in Portuguese).

Pérez J.-C. and Vidal E. (1992). "An Algorithm for the Optimum Piecewise Linear Approximation of Digitized Curves", *In Proc. International Conference on Pattern Recognition – ICPR'92*, pp. 167-170.

Pfeiffer S., Fischer S. and Effelsberg W. (1996). "Automatic Audio Content Analysis", *In Proc. ACM Multimedia Conference – Multimedia'96*, pp. 21-30.

Pikrakis A. and Theodoridis S. (2005). "A Novel HMM Approach to Melody Spotting in Raw Audio Recordings", *In Proc. International Conference on Music Information Retrieval – ISMIR'2005*.

Piszczalski M. and Galler B. A. (1977). "Automatic Music Transcription", *Computer Music Journal*, Vol. 1, No. 4, pp. 24-31.

Plumbey M. D., Abdallah S. A., Bello J. P., Davies M. E., Klingseisen J., Monti G. and Sandler M. B. (2001). "ICA and Related Models Applied to Audio Analysis and Separation", *In Proc. International ICSC Symposium on Soft Computing and Intelligent Systems for Industry – SOCO/ISFI'2001*.

Polikar R. (1999). *The Wavelet Tutorial – Part II, Fundamentals: The Fourier Transform and the Short Time Fourier Transform*, Technical Report, College of Engineering, Rowan

University, URL: http://users.rowan.edu/~polikar/WAVELETS/WTpart2.html, available by July 31, 2006.

Poliner G. and Ellis D. (2005a). "A Classification Approach to Melody Transcription", *In Proc. International Conference on Music Information Retrieval – ISMIR'2005.*

Poliner G. and Ellis D. (2005b). "A Classification Approach to Melody Transcription", *In Proc. Music Information Retrieval Exchange – MIREX'2005.*

Rabiner L. and Juang B. H. (1993). *Fundamentals of Speech Recognition*, Prentice-Hall.

Rolland P. Y., Raskinis G. and Ganascia J. G. (1999). "Musical Content-based Retrieval: An Overview of the Melodiscov Approach and System", *In Proc. ACM Multimedia Conference – Multimedia'99*, pp. 81-84.

Ryynänen M. P. (2004). *Probabilistic Modelling of Note Events in the Transcription of Monophonic Melodies*, MSc Thesis, Tampere University of Technology, Finland.

Ryynänen M. P. and Klapuri A. (2005a). "Polyphonic Music Transcription Using Note Event Modeling", *In Proc. IEEE Workshop on Applications of Signal Processing to Audio and Acoustics – WASPAA'2005.*

Ryynänen M. P. and Klapuri A. (2005b). "Note Event Modeling for Audio Melody Extraction", *In Proc. Music Information Retrieval Exchange – MIREX'2005.*

Scheirer E. D. (1995). "Using Musical Knowledge to Extract Expressive Performance Information from Audio Recordings", *In Proc. International Joint Conference on Artificial Intelligence – IJCAI'95, Workshop on Computational Auditory Scene Analysis*, pp. 153-160.

Scheirer E. D. (1998). "Tempo and Beat Analysis of Acoustic Musical Signals", *Journal of the Acoustical Society of America*, Vol. 103, No. 1, pp. 588-601.

Scheirer E. D. (2000). *Music-Listening Systems*, PhD Thesis, School of Architecture and Planning, Massachusetts Institute of Technology, USA.

Selfridge-Field E. (1998). "Conceptual and Representational Issues in Melodic Comparison", *Computing in Musicology*, Vol. 11, pp. 3-64.

Serra X. (1989). *A System for Sound Analysis/Transformation/Synthesis Based on a Deterministic Plus Stochastic Decomposition*, PhD Thesis, Department of Music, Stanford University, USA.

Serra X. (1997). "Musical Sound Modeling with Sinusoids Plus Noise", In Roads C., Pope S., Picialli A. and De Poli G. (eds.): *Musical Signal Processing*, Swets & Zeitlinger Publishers.

Shih H.-H., Narayanan S. S. and Kuo C.-C. J. (2003). "Multidimensional Humming Transcription Using a Statistical Approach for Query by Humming Systems", *In Proc. IEEE International Conference on Acoustics Speech and Signal Processing – ICASSP'2003.*

Slaney M. (1988). *Lyon's Cochlear Model*, Apple Computer Technical Report #13.

Slaney M. (1998). *Auditory Toolbox: A Matlab Toolbox for Auditory Modeling Work (version 2)*, Technical Report, Interval Research Corporation, Palo Alto, USA.

Slaney M. and Lyon R. F. (1990). "A Perceptual Pitch Detector", *In Proc. IEEE International Conference on Acoustics Speech and Signal Processing – ICASSP'90*.

Slaney M. and Lyon R. F. (1993). "On the Importance of Time - A Temporal Representation of Sound", In Cooke M., Beet S. and Crawford M. (eds.): *Visual Representations of Speech Signals*, John Wiley & Sons.

Smaragdis P. (2001). *Redundancy Reduction for Computational Audition, a Unifying Approach*, PhD Thesis, School of Architecture and Planning, Massachusetts Institute of Technology, USA.

Smith S. W. (1997). *The Scientist and Engineer's Guide to Digital Signal Processing*, California Technical Publishing.

Song J., Bae S. Y. and Yoon K. (2002). "Mid-Level Music Melody Representation of Polyphonic Audio for Query-by-Humming System", *In Proc. International Conference on Music Information Retrieval – ISMIR'2002*.

Sterian A. D. (1999). *Model-based Segmentation of Time-Frequency Images for Music Transcription*, PhD Thesis, Department of Electrical Engineering and Computer Science, University of Michigan, USA.

Talkin D. (1995). "A Robust Algorithm for Pitch Tracking", In Kleijn W. B and Paliwal K. K. (eds.): *Speech Coding and Synthesis*, pp. 495-518, John Wiley & Sons.

TechWhack (2005). "Apple iTunes Touches an Impressive 250 Million Sales Figure", *TechWhack.com*, URL: http://news.techwhack.com/690/apple-itunes-250-million/, published in January 25, 2005, available by July 31, 2006.

Temperley D. (2001). *Cognition of Basic Musical Structures*, MIT Press.

Tolonen T. and Karjalainen M. (2000). "A Computationally Efficient Multipitch Analysis Model", *IEEE Transactions on Speech and Audio Processing*, Vol. 8, No. 6, pp. 708-716.

Tsur R. (2000). "Metaphor and Figure-Ground Relationship: Comparisons from Poetry, Music, and the Visual Arts", *PsyArt - An Online Journal for the Psychological Study of the Arts*, 2000 issue, URL: http://www.clas.ufl.edu/ipsa/journal/2000_tsur03.shtml.

Tzanetakis G. (2002). *Manipulation, Analysis and Retrieval Systems for Audio Signals*, PhD Thesis, Department of Computer Science, Princeton University, USA.

Uitdenbogerd A. L., Chattaraj A. and Zobel J. (2000). "Music IR: Past, Present and Future", *In Proc. International Symposium on Music Information Retrieval – ISMIR'2000*.

Uitdenbogerd A. L. (2002). *Music Information Retrieval Technology*, PhD Thesis, Department of Computer Science, RMIT University, Melbourne, Australia.

Vembu S. and Baumann S. (2004). "A Self-Organizing Map Based Knowledge Discovery for Music Recommendation Systems", *In Proc. International Symposium on Computer Music Modeling and Retrieval – CMMR'2004*.

Vignoli F. and Pauws S. (2005). "A Music Retrieval System Based on User-driven Similarity and Its Evaluation", *In Proc. International Conference on Music Information Retrieval – ISMIR'2005*.

Viitaniemi T., Klapuri A. and Eronen A. (2003). "A Probabilistic Model for the Transcription of Single-Voice Melodies", *In Proc. Finnish Signal Processing Symposium – FINSIG'2003*.

Vincent E. and Plumbey M. D. (2005). "Predominant-F0 Estimation Using Bayesian Harmonic Waveform Models", *In Proc. Music Information Retrieval Exchange – MIREX'2005*.

Vincent E. and Rodet X. (2004). "Instrument Identification in Solo and Ensemble Music Using Independent Subspace Analysis", *In Proc. International Conference on Music Information Retrieval – ISMIR'2004*.

Virtanen T. (2003). "Sound Source Separation Using Sparse Coding with Temporal Continuity Objective", *In Proc. International Computer Music Conference – ICMC'2003*.

Virtanen T. and Klapuri A. (2000). "Separation of Harmonic Sound Sources Using Sinusoidal Modeling", *In Proc. IEEE International Conference on Acoustics, Speech and Signal Processing – ICASSP'2000*.

Wagner R. A. and Fischer M. J. (1974). "The String-to-String Correction Problem", *Journal of the* ACM, Vol. 21, No. 1, pp. 168-173.

Wang Y., Liu Z. and Huang J.-C. (2000). "Multimedia Content Analysis Using Both Audio and Visual Clues", *IEEE Signal Processing Magazine*, Vol. 17, No. 6, pp. 12-36.

Welsh M., Borisov N., Hill J., von Behren R. and Woo A. (1999). *Querying Large Collections of Music for Similarity*, Technical Report, Computer Science Division, University of California, Berkeley, USA.

Wold E., Blum T., Keislar D. and Wheaton J. (1996). "Content-based Classification, Search and Retrieval of Audio", *IEEE Multimedia*, Vol. 3, No. 3, pp. 27-36.

Yang C. (2001). "Music Database Retrieval Based on Spectral Similarity", *In Proc. International Symposium on Music Information Retrieval – ISMIR'2001*.

APPENDIX A

OTHER EVALUATED PITCH DETECTION APPROACHES

Besides the AMPD, we also evaluated other different kinds of approaches, based on spectral, autocorrelation, spectral autocorrelation and probabilistic analyses. All of them conform to the general framework presented in Section 3.3:

i) selection of a fixed analysis frame;

ii) definition of some sort of pitch salience curve in each frame (e.g., summary correlogram, autocorrelation function, energy associated with different F0 candidates, spectral ACF or probabilistic likelihood of each possible F0);

iii) peak detection in the pitch salience curve;

iv) selection of the most salient pitch candidates (in a maximum of *maxNPC*, i.e., 5 in our implementation).

Both temporal and spectral autocorrelation were previously described. Thus, only an algorithm devised by us, relying on STFT-based harmonic analysis, and a probabilistic approach, proposed in [Goto, 2000], are described here.

A.1. STFT-based Harmonic Analysis

The algorithm described in the following paragraphs was our first attempt towards melody-oriented pitch detection, based on the fact that the Short-Time Fourier Transform is one of the most widely used time-frequency analysis technique.

Briefly, the Discrete Fourier Transform (DFT) provides information about how much of each frequency is present in a signal. This works well for static signals, for which the spectral content of the sound does not change significantly over time. However, in musical signals, notoriously time-varying in nature, the Fourier transform is unable to distinguish the different frequencies present. Instead, it shows information regarding all

the existing frequencies, regardless of their occurrence in time.

The STFT is then suggested as an attempt to deal with the lack of time resolution in the DFT. To this end, the input signal is divided into small sequential frames of analysis for which (quasi-)stationarity can be assumed (46.44 msec in our case). Then, the standard DFT is applied to each of these frames in succession. The result is a time-dependent representation, showing the changes in the frequency spectrum as the signal progresses. A comprehensive overview of the DFT and the STFT can be found in [Smith, 1997; Polikar, 1999].

A. Windowing

The standard DFT assumes a signal of theoretically infinite length. In order to cope with finite-length signals, these are expanded to infinite length by repeating them an infinite number of times. As a consequence, in STFT analysis a discontinuity or break in the signal occurs at frame boundaries. As a result, spurious spectral components appear. Indeed, a simple division of the signal into frames is the same as multiplying it by a sliding rectangular window, characterized by its great amount of spectral leakage.

This problem is usually tackled by applying a windowing function to the frame, which smoothly scales the amplitude of the signal to zero at each border, reducing the discontinuities at frame boundaries. Thus, windowing has the advantage of reducing the presence of spurious spectral components. As before, we use a Hamming window, which has proved to offer a good trade-off between spectral leakage and spectral resolution, besides being simple to implement and computationally efficient [Smith, 1997, pp. 286].

When a window is applied to a signal, some information near the frame boundaries is obviously lost. For this reason, the STFT is further improved by imposing some overlapping between consecutive frames. In this way, information that is lost in a frame one is picked up in another. A hop size of 5.8 msec was defined, as previously.

B. Zero-padding

In order to reduce the spectral frequency intervals, each frame is zero-padded. Zero-padding does not improve resolution but improves single peak location accuracy, which is important for acquiring more accurate peak frequencies. Furthermore, the DFT is performed more efficiently with the FFT algorithm, which is optimized for speed when the number of samples is a power of 2. Therefore, we added the number of zeros that is necessary to attain 4096 samples, originating a frequency interval of 5.38 Hz, which seems adequate. In effect, the melody is usually in a mid-range frequency, above 100 Hz. Since the frequency difference between notes A2 (110 Hz) and A2# (116.54 Hz) is above our threshold, the defined interval looks appropriate. Peak location accuracy can be further improved by peak interpolation [Martins, 2001, pp. 32; Serra, 1989, pp. 43].

C. Evaluation of the Magnitude Spectrum

After defining a windowed, zero-padded signal frame, its magnitude spectrum is achieved via the FFT. Additionally, we convert the spectrum to dB units, taking its logarithm. The reason for doing this is that we found experimentally that spectral peaks show up more clearly in the logarithmic magnitude spectrum. Formally, it comes (A.1):

$$X = 20\log_{10}\left|FFT\left(x_{wz}\right)\right| \tag{A.1}$$

where x_{wz} denotes the windowed zero-padded frame signal and X represents the magnitude spectrum in dB.

D. Detection of F0 Candidates

Next, we look for peaks in the magnitude spectrum, according to the assumption that the fundamental frequencies present in the signal correspond to clear peaks in the spectrum. We look for all local maxima rather than only prominent peaks, for the previously described reasons (Section 3.3.4).

After detecting all spectral peaks, we identify a set of candidate harmonic groups, found by grouping together harmonically-related peaks (in a similar way to [Martins, 2001]). These groups are characterized by their F0s and energies. We start by finding the highest spectral peak, which is our first F0 candidate. Then, for each candidate *ff* (determined as explained in the following paragraph), we find all its harmonic candidates. A given frequency peak is considered an harmonic of a candidate peak if its frequency deviates at most 50 cents from the theoretical harmonic value, i.e., is in the range (A.2):

$$\left[\frac{k \cdot ff}{r};\ (k \cdot ff) \cdot r\right],\ r = 2^{\frac{50}{1200}} \tag{A.2}$$

In (A.2), r is the ratio for obtaining the frequency range k stands for the k^{th} harmonic of the fundamental frequency *ff*.

Not all the detected peaks are F0 candidates. In reality, we define a threshold for the minimum peak amplitude such that F0 peak candidates might not differ by more than 35 dB from the maximum peak found.

Once we have found all the harmonic groups, their respective energies are computed by summing up the amplitudes of the peaks belonging to each group. Since we took the logarithmic magnitude of the spectrum, we convert peak magnitudes back to their original values by inverting the logarithm. After that, only the groups with sufficient energy are kept. To accomplish this, we calculate the maximum group energy and determine the minimum allowed group energy using the minimum energy ratio parameter, *minSalRatio*, as in Chapter 3. We then eliminate all groups whose energies are below this threshold.

Finally, the energies of the harmonic groups in all frames are normalized to the [0; 100] interval.

Clearly, one of the main drawbacks of this approach is that no F0 will be detected in the case of a masked or missing first harmonic. This is particularly problematic given the fact that, in typical popular music, the harmonic structure of the leading soloist (e.g., singing voice) is often overlapped by the higher harmonics of the bass or masked by percussive sounds. Therefore, another approach, e.g., making use of comb filtering, could be followed with the same purpose.

The results of this method are illustrated in Figure A.1 for our saxophone riff, where the candidate F0s are circled (bottom panel). The harmonic group with the highest energy corresponds to the peak at 371.4 Hz, being very close to the real pitch value (at about 370 Hz). In addition, super-harmonics are also output as F0 candidates.

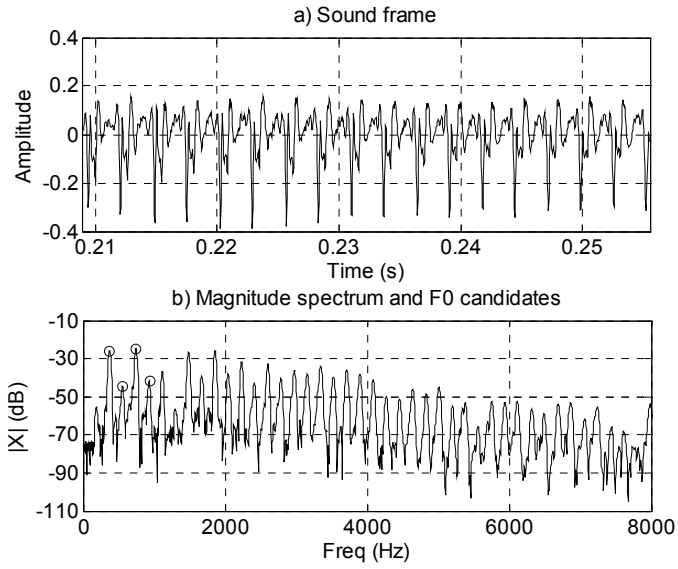


Figure A.1. Results of the STFT-based harmonic analysis.

A.2. Probabilistic Approach

We are also interested in evaluating a probabilistic approach, where pitch likelihoods are calculated from the magnitude spectrum. To this end, we based ourselves on the algorithm described in [Goto, 2000], with some adaptations. Basically, the method regards the observed frequency components as a weighted mixture of harmonic-structure tone

models and estimates their weights by using the expectation-maximization algorithm. These weights represent F0 likelihoods.

A. Spectrum Evaluation

The author started by designing an STFT-based multirate filterbank that separates the signals in five bands, namely: 0 – 450, 450 – 900, 900 – 1800, 1800 – 3600 and 3600 - 7200 Hz. The instantaneous frequency, i.e., the rate of change of the phase of the signal - based on the phase vocoder [Flanagan and Golden, 1966] - is then computed. Afterwards, candidate frequency components are extracted by means of a frequency-to-instantaneous-frequency mapping and the magnitude spectrum in the corresponding frequencies is obtained.

However, in our implementation, no multirate filterbank was employed and no candidate frequencies were determined. Rather, we directly used the magnitude spectrum for all frequencies from zero to Nyquist, as in the previous section. Thus, all frequency bins are used as F0 candidates.

B. Frequency Region of Analysis

One of the main assumptions in this model is that the main melodic line has the most important harmonic structure in middle and high frequency regions. Namely, a band-pass filter is designed so that it covers most of the dominant harmonics of typical melodies and de-emphasizes crowded frequency regions around the F0. No matter if the F0 is within that range or not, the method attempts its estimation with recourse to the frequency components in the defined range that support it.

The characteristics of the BPF filter are best explained in a logarithmic frequency scale. Namely, the logarithmic scale used in equal temperament tuning is defined. Hence, frequency values in Hz units (linear scale) are converted to cents (logarithmic scale), as described in Section 2.6.2.

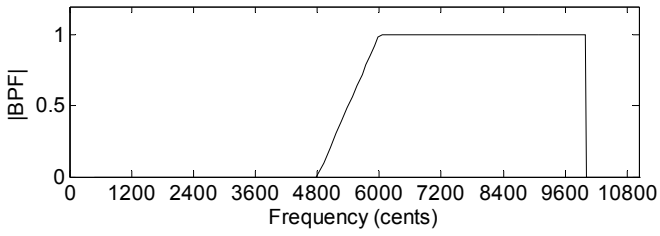


Figure A.2. Frequency response of the melodic band-pass filter.

The frequency response of the BPF spans a range from 4800 to slightly above 9600 cents (i.e., 261.6 Hz to 4186 Hz). In our implementation, we defined a maximum of 10000 cents, i.e., 5274 Hz. The transition band is linear (in the logarithmic scale), from 4800 to 6000 cents (523.3 Hz). The overall frequency response is sketched in Figure A.2.

C. Calculation of the F0's Probability Density Function

The observed probability density function of the band-pass filtered frequency components, p_X, in a given time frame is defined as in (A.3):

$$\begin{aligned} \mathcal{X}_{BPF}(f) &= BPF(f) \cdot \left|\mathcal{X}(f)\right| \\ p_{\mathcal{X}}(f) &= \frac{\mathcal{X}_{BPF}(f)}{\mathrm{power}\left(\mathcal{X}_{BPF}(f)\right)} \end{aligned} \tag{A.3}$$

where $BPF(f)$[62] denotes the frequency response of the band-pass filter and $\mathcal{X}(f)$ stands for the spectrum of the waveform in a given analysis frame, at frequency f.

In order to obtain the PDF of the F0, the basic idea is to consider that the observed PDF, p_X, was generated from a model that is a weighted mixture of harmonic-structure tone models. The PDF of a tone model, $p(f|F)$, which indicates where the harmonics of the fundamental frequency F tend to occur, is defined by Gaussian distributions centered at the theoretical locations of the harmonics, according to (A.4):

$$\begin{aligned} p(f \mid F) &= \alpha \sum_{h=1}^{N} c(h)\, \mathrm{G}\left(f; F + 1200 \cdot \log_2 h, W\right) \\ \mathrm{G}(f; m, \sigma) &= \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(f-m)^2}{2\sigma^2}} \\ c(h) &= \mathrm{G}(h; 1, H) \end{aligned} \tag{A.4}$$

In the previous expression, α is a normalization factor used to guarantee that the PDF's integral equals 1, $N = 16$ is the number of harmonics considered, $W = 17$ cents is the standard deviation of the Gaussian distribution $G(f; m, \sigma)$ and $c(h)$ specifies the amplitude of the h^{th} harmonic, also defined as a Gaussian with unity mean and standard deviation $H = 5.5$[63]. This is the model presented in [Goto, 2000], which was later extended in [Goto, 2001] in order to incorporate adaptive tone models. However, since the average results did not improve substantially (about 1.9%), we kept the original model for simplicity.

---

[62] BPF, $\mathcal{X}$, $\mathcal{X}_{BPF}$ and p_X are all discrete entities. However, we use (.) rather than [.] for the sake of uniformity with the authors continuous notation.

[63] We use the default parameters specified by the author, except when explicitly stated.

Next, we define a mixture density, $p(f; \theta)$, as a weighted combination of the harmonic-structure tone models. Formally, it comes (A.5):

$$p(f;\theta) = \int_{F_l}^{F_h} w(F) \cdot p(f \mid F) dF$$
$$\theta = \{w(F) : F_l \le F \le F_h\} \tag{A.5}$$

There, F_l and F_h denote the lower and upper bounds of the allowable F0 range (3600 and 9600 cents, respectively, i.e., 130.8 and 4186 Hz). In our implementation, the integral is computed using a frequency interval of 20 cents. In (A.5), $w(F)$ is the weight of the tone model with F0 = F. The distribution of weights must satisfy (A.6):

$$\int_{F_l}^{F_h} w(F) \cdot dF = 1 \tag{A.6}$$

The problem is then to estimate the parameter θ (.e., the weights of the tone models) so that the observed PDF, p_X, is likely to have been generated from the model $p(f; \theta)$, where the weights, $w(F)$, can be interpreted as the PDF of the F0. Naturally, the weight of a particular tone model will be strongly correlated to the energy of their observed harmonics.

To this end, the maximum likelihood estimation of θ, θ_{ML}, is achieved by maximizing the mean log-likelihood defined as (A.7):

$$\theta_{ML} = \arg\max_{\theta} (MLL)$$
$$MLL = \int_{-\infty}^{+\infty} p_X(f) \cdot \log(p(f;\theta)) df \tag{A.7}$$

This is accomplished iteratively with recourse to the expectation-maximization algorithm. Details regarding the proposed solution for this particular problem can be found in [Goto, 2000]. The author does not specify how the weights are initialized, neither the stopping criteria of the algorithm. Therefore, we assume uniform weight initialization and define as stopping conditions the stabilization of the conditional expectation of the mean likelihood, i.e., consecutive values differ by less than 0.01, or the fulfillment of a maximum of 20 iterations (values up to 10000 were experimented with similar results). Then, the weights for the following frame are start with the final weights in the previous one, as recommended by the author.

The obtained F0 PDF serves then as our pitch salience function, from where the highest peaks are selected as pitch candidates.

The results of this method are illustrated in Figure A.3 for a frame of the same saxo-

phone riff we have been using. The candidate F0s are signaled by circles in the bottom panel. There, the highest peak at 370.0 Hz corresponds to the true pitch.

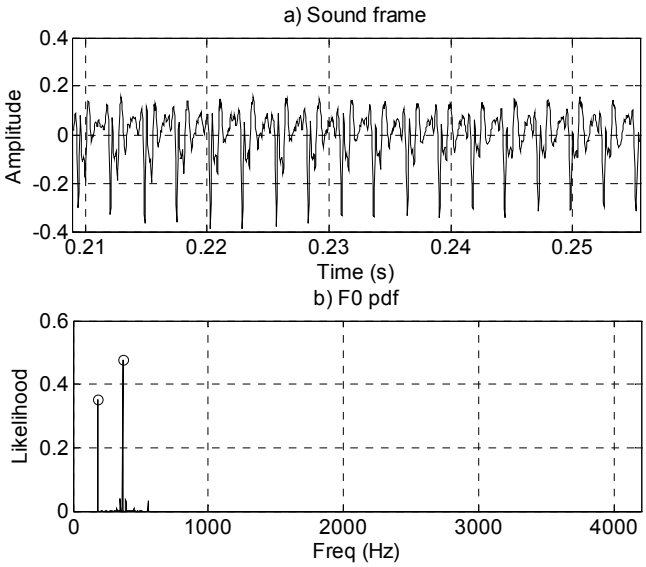


Figure A.3. Results of the probabilistic pitch detector.

APPENDIX B

DESCRIPTION OF SONG EXCERPTS

Further details on the used musical excerpts are provided here. These, as well as annotation and result files, can be downloaded from http://www.dei.uc.pt/~ruipedro/MelodyDetection/. Some of the derived characterizations are qualitative and subjective.

1) Pachelbel's "Kanon"

Genre	Classical
Solo Type	Instrumental (MIDI synthesized)
Polyphonic Complexity	Low
SNR	High
Peculiarities	- Periodic abrupt note transitions - Strong bass
Duration	6.0 sec
# Melodic Notes	16

2) Handel's "Hallelujah"

Genre	Choral
Solo Type	Female vocal (choral soprano)
Polyphonic Complexity	High
SNR	Medium/Low
Peculiarities	- Counterpoint (four simultaneous singing voices) - Instrumental accompaniment in the foreground when the choir stops
Duration	5.54 sec
# Melodic Notes	15

3) Enya – "Only Time"

Genre	New Age
Solo Type	Female Vocal
Polyphonic Complexity	Low
SNR	High
Peculiarities	- Significant reverberant effects - Consecutive notes at the same pitch
Duration	5.95 sec
# Melodic Notes	11

4) Dido – "Thank You"

Genre	Pop
Solo Type	Female Vocal
Polyphonic Complexity	Medium
SNR	Medium/High (low when drums are hit)
Peculiarities	- Periodic strong percussive beats - Some glissando - Instrumental accompaniment in the foreground during melody pauses
Duration	6.0 sec
# Melodic Notes	16

5) Ricky Martin – "Private Emotion"

Genre	Pop
Solo Type	Male Vocal
Polyphonic Complexity	Medium
SNR	Medium (low when guitar strumming is very intense)
Peculiarities	- Intense guitar strumming and ambient background, which goes to the foreground during melody pauses - Some glissando
Duration	6.0 sec
# Melodic Notes	10

6) Avril Lavigne – "Complicated"

Genre	Pop/Rock
Solo Type	Female Vocal
Polyphonic Complexity	Medium
SNR	Medium (low when drums are hit)
Peculiarities	- Periodic strong percussive beats - Some glissando
Duration	4.27 sec
# Melodic Notes	14

7) Claudio Roditi – "Rua Dona Margarida"

Genre	Jazz/Easy
Solo Type	Instrumental (trumpet)
Polyphonic Complexity	Low
SNR	High
Peculiarities	- Periodic soft percussive beats - Harmony with a second background instrument in a lower pitch register - Consecutive notes at the same pitch
Duration	6.0 sec
# Melodic Notes	19

8) Mambo Kings – "Bella Maria de Mi Alma"

Genre	Bolero
Solo Type	Instrumental (trumpet)
Polyphonic Complexity	Low/Medium
SNR	High
Peculiarities	- Soft percussion - Some glissando
Duration	6.0 sec
# Melodic Notes	12

9) Eliades Ochoa – "Chan Chan"

Genre	Son
Solo Type	Male vocal
Polyphonic Complexity	High
SNR	Medium
Peculiarities	- Off-key singing and considerable glissando - Harmony with a second vocal in a lower pitch register and several latin percussive instruments - Consecutive notes at the same pitch
Duration	3.05 sec
# Melodic Notes	10

10) Juan Luis Guerra – "Palomita Blanca"

Genre	Bachata
Solo Type	Male vocal
Polyphonic Complexity	High
SNR	Medium
Peculiarities	- Long time intervals where the solo is absent - Several latin percussive instruments - Glissando and consecutive notes at the same pitch
Duration	6.0 sec
# Melodic Notes	11

11) Battlefield Band – "Snow on the Hills"

Genre	Scottish Folk
Solo Type	Instrumental (flute)
Polyphonic Complexity	Low
SNR	High
Peculiarities	- Flute and fiddle playing unison - Harmony with another (lower) accompanying instrument
Duration	5.84 sec
# Melodic Notes	26

12) daisy2

Genre	Pop
Solo Type	Female vocal (synthesized singing voice)
Polyphonic Complexity	Low
SNR	High
Peculiarities	- Unison (or one octave above) accompanying instrument - Soft ambient accompaniment
Duration	22.0 sec
# Melodic Notes	23

13) daisy3

Genre	Pop
Solo Type	Female vocal (synthesized singing voice)
Polyphonic Complexity	Low
SNR	High (sometimes low, when guitar strumming is intense)
Peculiarities	- Guitar strumming, sometimes more intense than the solo - Simple singing syllables
Duration	17.03 sec
# Melodic Notes	11

14) jazz2

Genre	Jazz
Solo Type	Instrumental (saxophone)
Polyphonic Complexity	Medium/Low
SNR	Medium
Peculiarities	- Soft solo, accompaniment and percussion - Instrumental accompaniment in the foreground during melody pauses - Consecutive notes at the same pitch
Duration	15.45 sec
# Melodic Notes	22

15) jazz3

Genre	Jazz
Solo Type	Instrumental (saxophone)
Polyphonic Complexity	Medium/High
SNR	Medium/High
Peculiarities	- Long time intervals where the solo is absent - Soft percussion
Duration	14.83 sec
# Melodic Notes	22

16) midi1

Genre	Pop
Solo Type	Instrumental (MIDI synthesized)
Polyphonic Complexity	Low
SNR	Medium/High
Peculiarities	- Soft solo, instrumental accompaniment and shakers
Duration	19.23 sec
# Melodic Notes	39

17) midi2

Genre	Folk
Solo Type	Instrumental (MIDI synthesized)
Polyphonic Complexity	Low
SNR	Medium/High (sometimes low due the harmony with a second part)
Peculiarities	- Harmony with second instrument - Consecutive notes at the same pitch
Duration	16.62 sec
# Melodic Notes	22

18) opera female 2

Genre	Opera
Solo Type	Female vocal
Polyphonic Complexity	Low
SNR	High
Peculiarities	- Extreme vibrato - Instrumental accompaniment in the foreground during melody pauses
Duration	16.11 sec
# Melodic Notes	37

19) opera male 3

Genre	Opera
Solo Type	Male vocal
Polyphonic Complexity	Low
SNR	High
Peculiarities	- Fast succession of short notes - "Triangular" melodic contour - Strong vibrato - Harmony with accompanying orchestral instruments
Duration	20.0 sec
# Melodic Notes	61

20) pop1

Genre	Pop
Solo Type	Male vocal
Polyphonic Complexity	Medium
SNR	Medium/High (low when second vocal is present)
Peculiarities	- Harmony with second vocal in a lower pitch register - Accompaniment is often more intense than the solo
Duration	22.71 sec
# Melodic Notes	34

21) pop4

Genre	Pop
Solo Type	Male vocal
Polyphonic Complexity	Medium
SNR	Medium/High
Peculiarities	- Fast beating - Some glissando - Harmony with a background instrument
Duration	21.66 sec
# Melodic Notes	29