

Ph.D. thesis

**Computationally efficient methods
for polyphonic music transcription**

Antonio Pertusa Ibáñez

Supervised by
José Manuel Iñesta Quereda



Universitat d'Alacant
Universidad de Alicante

Departament de Llenguatges i Sistemes Informàtics
Departamento de Lenguajes y Sistemas Informáticos

External reviewers:

Anssi Klapuri (Queen Mary University, London, UK)
Andreas Rauber (Vienna University of Technology, Austria)

Committee members:

Xavier Serra (Universitat Pompeu Fabra, Barcelona, Spain)
G rard Assayag (IRCAM, Paris, France)
Anssi Klapuri (Queen Mary University, London, UK)
Jos  Oncina (Universidad de Alicante, Spain)
Isabel Barbancho (Universidad de M laga, Spain)

A Teima

Acknowledgments

First and foremost, I would like to thank all members of the computer music lab from the University of Alicante for providing an excellent, inspiring, and pleasant working atmosphere. Especially, to the head of the group and supervisor of this work, Prof. José Manuel Iñesta. His encouraging scientific spirit provides an excellent framework for inspiring the new ideas that make us to continuously grow and advance. I own this work to his advice, support and help.

Carrying out a PhD is not an easy task without the help of so many people. First, I would like to thank all the wonderful staff of our GRFIA group, and in general, all the DLSI department from the University of Alicante. My research periods at the Audio Research Group from the Tampere University of Technology, the Music Technology Group from the Universitat Pompeu Fabra, and the Department of Software Technology and Interactive Systems from the Vienna University of Technology, also contributed decisively to make this work possible. I have learned much, as a scientist and as a person, from the wonderful and nice researchers of all these labs.

I would also thank to the people who directly contributed to this work. I am grateful to Dr. Francisco Moreno for delaying some of my teaching responsibilities when this work was in progress, and for supplying the k NN algorithms code. I learned most of the signal processing techniques needed for music transcription from Prof. Anssi Klapuri. I'll always be very grateful for the great period in Tampere and his kind hosting. He directly contributed to this dissertation providing the basis for the sinusoidal likeness measure code, and also the multiple f_0 databases that allowed to evaluate and improve the proposed algorithms. Thanks must also go to one of my undergraduate students, Jasón Box, which collaborated to this work building the ODB database and migrating the onset detection code from C++ into D2K.

I wish to express my gratitude to the referees of this dissertation, for kindly accepting the review process, and to the committee members.

This work would not have been possible without the primary support provided by the Spanish PROSEMUS project¹ and the Consolider Ingenio 2010 MIPRCV research program². It has also been funded by the Spanish CICYT projects TAR³ and TIRIG⁴, and partially supported by European Union-FEDER funds and the Generalitat Valenciana projects GV04B-541 and GV06/166.

Beyond research, I would like to thank my family and my friends (too many to list here, you know who you are). Although they don't exactly know what

¹Code TIN2006-14932-C02

²Code CSD2007-00018

³Code TIC2000-1703-CO3-02

⁴Code TIC2003-08496-C04

I am working on and will never read a boring technical report in English, their permanent understanding and friendship have actively contributed to keep my mind alive within this period.

Finally, this dissertation is dedicated to the most important person in my life, Teima, for her love, support, care and patience during this period.

Antonio Pertusa Ibáñez
February, 2010

Contents

1	Introduction	1
2	Background	7
2.1	Analysis of audio signals	7
2.1.1	Fourier transform	8
2.1.2	Time-frequency representations	11
2.1.3	Filters in the frequency domain	15
2.2	Analysis of musical signals	17
2.2.1	Dynamics	18
2.2.2	Timbre	19
2.2.3	Taxonomy of musical instruments	20
2.2.4	Pitched musical sounds	21
2.2.5	Unpitched musical sounds	24
2.2.6	Singing sounds	26
2.3	Music background	26
2.3.1	Tonal structure	27
2.3.2	Rhythm	31
2.3.3	Modern music notation	33
2.3.4	Computer music notation	34
2.4	Supervised learning	38
2.4.1	Neural networks	38
2.4.2	Nearest neighbors	40
3	Music transcription	43
3.1	Human music transcription	43
3.2	Multiple fundamental frequency estimation	45
3.2.1	Harmonic overlap	46
3.2.2	Beating	47
3.2.3	Evaluation metrics	48
3.3	Onset detection	52
3.3.1	Evaluation metrics	53
4	State of the art	55
4.1	Single fundamental frequency estimation	55
4.1.1	Time domain methods	55
4.1.2	Frequency domain methods	56
4.1.3	Perceptual models	59
4.1.4	Probabilistic models	60
4.2	Multiple fundamental frequency estimation	62
4.2.1	Salience methods	62

CONTENTS

4.2.2	Iterative cancellation methods	63
4.2.3	Joint estimation methods	65
4.2.4	Supervised learning methods	66
4.2.5	Unsupervised learning methods	69
4.2.6	Matching Pursuit methods	70
4.2.7	Bayesian models	72
4.2.8	Statistical spectral models	72
4.2.9	Blackboard systems	74
4.2.10	Database matching	75
4.3	Discussion of multiple f_0 estimation methods	76
4.4	Onset detection	77
4.4.1	Signal processing methods	77
4.4.2	Machine learning methods	81
4.5	Discussion of onset detection methods	82
5	Onset detection	83
5.1	Methodology	84
5.1.1	Preprocessing	84
5.1.2	Onset detection functions	85
5.1.3	Peak detection and thresholding	89
5.2	Evaluation with the ODB database	89
5.2.1	Results using $o[t]$	89
5.2.2	Results using $\hat{o}[t]$	92
5.3	MIREX evaluation	93
5.3.1	Methods submitted to MIREX 2009	93
5.3.2	MIREX 2009 onset detection results	95
5.4	Conclusions	98
6	Multiple pitch estimation using supervised learning	101
6.1	Preprocessing	102
6.1.1	Construction of the input-output pairs	102
6.2	Supervised methods	103
6.2.1	Time-delay neural networks	104
6.2.2	Nearest neighbors	105
6.3	Evaluation	106
6.3.1	Generation of the training and test sets	107
6.4	Results using time-delay neural networks	108
6.4.1	Neural network parametrization	108
6.4.2	Recognition results	109
6.4.3	Changing waveshapes for detection	111
6.5	Results using k nearest neighbors	113
6.6	Conclusions	114

7	Multiple f_0 estimation using signal processing methods	119
7.1	Iterative cancellation method	120
7.1.1	Preprocessing	120
7.1.2	Onset detection	123
7.1.3	Candidate selection	123
7.1.4	Iterative cancellation	123
7.1.5	Postprocessing	126
7.2	Joint estimation method I	126
7.2.1	Preprocessing	127
7.2.2	Candidate selection	128
7.2.3	Generation of combinations of candidates	129
7.2.4	HPS estimation	129
7.2.5	Saliency of a combination	131
7.2.6	Postprocessing	133
7.3	Joint estimation method II	133
7.3.1	Temporal smoothing	134
7.3.2	Partial search	136
7.3.3	Fundamental frequency tracking	138
7.3.4	Alternative architectures	139
7.4	Evaluation	141
7.4.1	Parametrization	141
7.4.2	Results using random mixtures	149
7.4.3	Evaluation and comparison with other methods	157
7.4.4	Overall MIREX comparison	162
7.5	Conclusions	168
8	Conclusions and future work	171
8.1	Discussion and future lines of work	172
8.2	Publications	174
A	Resumen	177
	Bibliography	191

List of Figures

1.1	Music transcription example	3
2.1	Complex plane diagram	9
2.2	Example spectrum	10
2.3	Example magnitude spectrogram	12
2.4	Wavelet filter bank	13
2.5	Time-frequency resolution grids	14
2.6	Example filter bank	15
2.7	Mel scale	17
2.8	Attack, sustain and release	18
2.9	Harmonic spectrum	21
2.10	Piano waveform and spectrogram	22
2.11	Vibraphone waveform and spectrogram	25
2.12	Snare waveform and spectrogram	25
2.13	Voice waveform and spectrogram	27
2.14	Western musical notes	28
2.15	Musical keys	29
2.16	Harmonics and intervals. Fig. from Krumhansl (2004)	30
2.17	Metrical levels and timing. Fig. from Hainsworth (2003)	32
2.18	Modern notation example	35
2.19	Note names, score location, frequencies and MIDI pitches	36
2.20	MIDI visual representations example	38
2.21	Multilayer perceptron architecture	39
2.22	TDNN architecture. Fig. from Duda et al. (2000)	40
2.23	Example of NN classification in a 2D feature space	41
3.1	Beating in the time domain	48
3.2	Beating in the frequency domain	49
3.3	Onset detection example	53
4.1	Maximum likelihood from Noll (1969)	58
4.2	Two way mismatch from Maher and Beauchamp (1994)	59
4.3	Note and musicological models from Ryynänen and Klapuri (2004)	61
4.4	Iterative cancellation method from Klapuri (2003a)	63
4.5	Probabilistic framework from Ryynänen and Klapuri (2005)	64
4.6	Overview of the joint estimation method from Yeh (2008)	65
4.7	SONIC scheme from Marolt (2004a)	67
4.8	HMM smoothed estimation from Poliner and Ellis (2007a)	68
4.9	NMF example from Smaragdis and Brown (2003)	70
4.10	Modified MP algorithm from Leveau et al. (2008)	71

LIST OF FIGURES

4.11	Overview of the system proposed by Goto (2000).	73
4.12	HTC spectral model of a single source from Kameoka et al. (2007)	74
4.13	Blackboard architecture from Martin (1996)	75
4.14	General onset detection scheme	78
5.1	One semitone filter bank	84
5.2	Example of the onset detection function for a piano melody	86
5.3	Onset detection function for a polyphonic violin song	88
5.4	SFS screenshot	90
5.5	Onset detection precision and recall	90
5.6	D2K itinerary of the onset detection system	94
5.7	MIREX onset detection results respect to θ	97
6.1	Binary digital piano-roll (BDP)	104
6.2	TDNN architecture and data supplied during training	105
6.3	Sigmoid transfer function	105
6.4	Sinusoidal waveform and spectrum	107
6.5	Sawtooth waveform and spectrum	107
6.6	Clarinet waveform and spectrum	108
6.7	Hammond waveform and spectrum	108
6.8	TDNN recognition accuracy as a function of pitch	110
6.9	TDDN accuracy respect to the amount of pitches	111
6.10	Example TDNN detection using a clarinet sound	112
6.11	Event detection accuracy using A_p for the sinusoidal timbre	114
6.12	Event detection accuracy using A'_p for the sinusoidal timbre	115
6.13	Event detection accuracy using A''_p for the sinusoidal timbre	115
7.1	Iterative cancellation scheme.	121
7.2	SLM example	122
7.3	Candidate cancellation example	125
7.4	Interpolation example	130
7.5	Spectral smoothness measure example	132
7.6	Example of combinations merged across adjacent frames	135
7.7	Example of candidate intensities for an oboe sound	136
7.8	Example of candidate intensities for a mixture	137
7.9	Partial selection in the joint estimation method II	138
7.10	wDAG example	139
7.11	SLM evaluation	143
7.12	Iterative cancellation accuracy adjusting the parameters	144
7.13	Joint method I candidate identification adjusting the parameters	146
7.14	Joint method I accuracy adjusting the parameters	146
7.15	Joint method I runtime adjusting the parameters	147

LIST OF FIGURES

7.16 Joint method II accuracy adjusting the parameters 149

7.17 Candidate identification results 150

7.18 Global candidate identification results 150

7.19 Pitch detection results for the iterative cancellation method . . . 151

7.20 Pitch detection results for the joint estimation method I 151

7.21 Pitch detection results for the joint estimation method II 152

7.22 Global pitch detection results 152

7.23 Polyphony estimation results using with one source 153

7.24 Polyphony estimation results with two simultaneous sources . . . 153

7.25 Polyphony estimation results with four simultaneous sources . . . 154

7.26 Polyphony estimation results with six simultaneous sources . . . 154

7.27 Global polyphony estimation results 155

7.28 Results of the iterative cancellation method respect to the pitch . 155

7.29 Results of the joint estimation method I respect to the pitch . . . 156

7.30 Results of the joint estimation method II in respect to the pitch . 156

7.31 MIREX 2007-2008 E_{tot} . Fig. from [Bay et al. \(2009\)](#) 165

7.32 MIREX 2007-2008 accuracy. Fig. from [Bay et al. \(2009\)](#) 166

7.33 MIREX 2007-2008 note tracking F-m. Fig. from [Bay et al. \(2009\)](#) 167

List of Tables

2.1	Symbols representing note durations	34
5.1	Onset detection results using the ODB database	91
5.2	Comparison with other works using the ODB database	92
5.3	Onset detection results using short context	93
5.4	Overall MIREX 2009 onset detection results	95
5.5	MIREX 2009 onset detection runtimes	96
5.6	MIREX 2009 onset detection results using the best θ	96
5.7	MIREX 2009 poly-pitched results	96
6.1	Frame-by-frame and note detection accuracy using TDNN	109
6.2	Frame level cross-detection results using TDNN	113
6.3	Note level cross-detection results using TDNN	113
6.4	Event accuracy for each activation function using k NN	116
6.5	Note accuracy for each activation function using k NN	116
7.1	Iterative cancellation method parameters	142
7.2	Joint estimation method I parameters	145
7.3	Joint estimation method II parameters	148
7.4	MIREX (2007) note tracking runtimes	158
7.5	MIREX (2007) note tracking results based on onset and pitch	159
7.6	MIREX (2007) frame by frame evaluation results	160
7.7	MIREX (2007) frame by frame runtimes	161
7.8	MIREX (2008) frame by frame runtimes	162
7.9	MIREX (2008) note tracking runtimes	162
7.10	MIREX (2008) frame by frame evaluation results	163
7.11	MIREX (2008) note tracking results	164

1

Introduction

Automatic music transcription is a music information retrieval (MIR) task which involves many different disciplines, such as audio signal processing, machine learning, computer science, psychoacoustics and music perception, music theory, and music cognition.

The goal of automatic music transcription is to extract a human readable and interpretable representation, like a musical score, from an audio signal. A score is a guide to perform a piece of music, and it can be represented in different ways. The most extended score representation is the modern notation used in Western tonal music. In order to extract a readable score from a signal, it is necessary to estimate the pitches, onset times and durations of the notes, the tempo, the meter and the tonality of a musical piece.

The most obvious application of automatic music transcription is to help a musician to write down the music notation of a performance from an audio recording, which is a time consuming task when it is done by hand. Besides this application, automatic music transcription can also be useful for other MIR tasks, like plagiarism detection, artist identification, genre classification, and composition assistance by changing the instrumentation, the arrangement or the loudness before resynthesizing new pieces. In general, music transcription methods can also provide information about the notes to symbolic music algorithms.

The transcription process can be separated into two main stages: to convert an audio signal to a piano-roll representation, and to convert the estimated piano-roll into musical notation.

As pointed out by [Cemgil et al. \(2003\)](#), most authors only consider automatic music transcription as an audio to piano-roll conversion, whereas piano-roll to score notation can be seen as a separate problem. This can be justified since the processes involved in audio to piano-roll notation include pitch estimation and temporal note segmentation, which constitutes a challenging task itself. The piano-roll to score process involves tasks like tempo estimation, rhythm

1. INTRODUCTION

quantization, key detection or pitch spelling. This stage is more related to the generation of human readable notation.

In general, a music transcription system can not obtain the exact score that the musician originally read. Musical audio signals are often expressive performances, rather than simple mechanical translations of notes read on a sheet. A particular score can be performed by a musician in different ways. As scores are only guides for the performers to play musical pieces, converting the notes present in an audio signal into staff notation is an ill-posed problem without a unique solution.

However, the conversion of a musical audio signal into a piano-roll representation without rhythmic information only depends on the waveform. Rather than a score-oriented representation, a piano-roll can be seen as a sound-oriented representation which displays all the notes that are playing at each time. The conversion from an audio file into a piano-roll representation is done by a multiple fundamental frequency (f_0) estimation method. This is the main module of a music transcription system, as it estimates the number of notes sounding at each time and their pitches.

For converting a piano-roll into a readable score, other harmonic and rhythmic components must also be taken into account. The tonality is related with the musical harmony, showing hierarchical pitch relationships based on a key tonic. Source separation and timbre classification can be used to identify the different instruments present in the signal, allowing the extraction of individual scores for each instrument. The metrical structure refers to the hierarchical temporal structure. It specifies how many beats are in each measure and what note value constitutes one beat, so bars can be added to the score to make it readable by a musician. The tempo is a measure to specify how fast or slow is a musical piece.

A music transcription example is shown in Fig. 1.1. The audio performance of the score in the top of the figure was synthesized for simplification, and it did contain neither temporal deviations nor pedal sustains. The piano-roll inference was done without errors. The key, tempo, and meter estimates can be inferred from the waveform or from the symbolic piano-roll representation. In this example, these estimates were also correct (except from the anacrusis¹ at the beginning, which causes the shift of all the bars). However, it can be seen that the resulting score differs from the original one.

When a musician performs a score, the problem is even more challenging, as there are frequent temporal deviations, and the onset and duration of the notes must be adjusted (quantized) to obtain a readable score. Note that quantizing temporal deviations implies that the synthesized waveform of the

¹The term anacrusis refers to the note or sequence of notes which precede the beginning of the first bar.

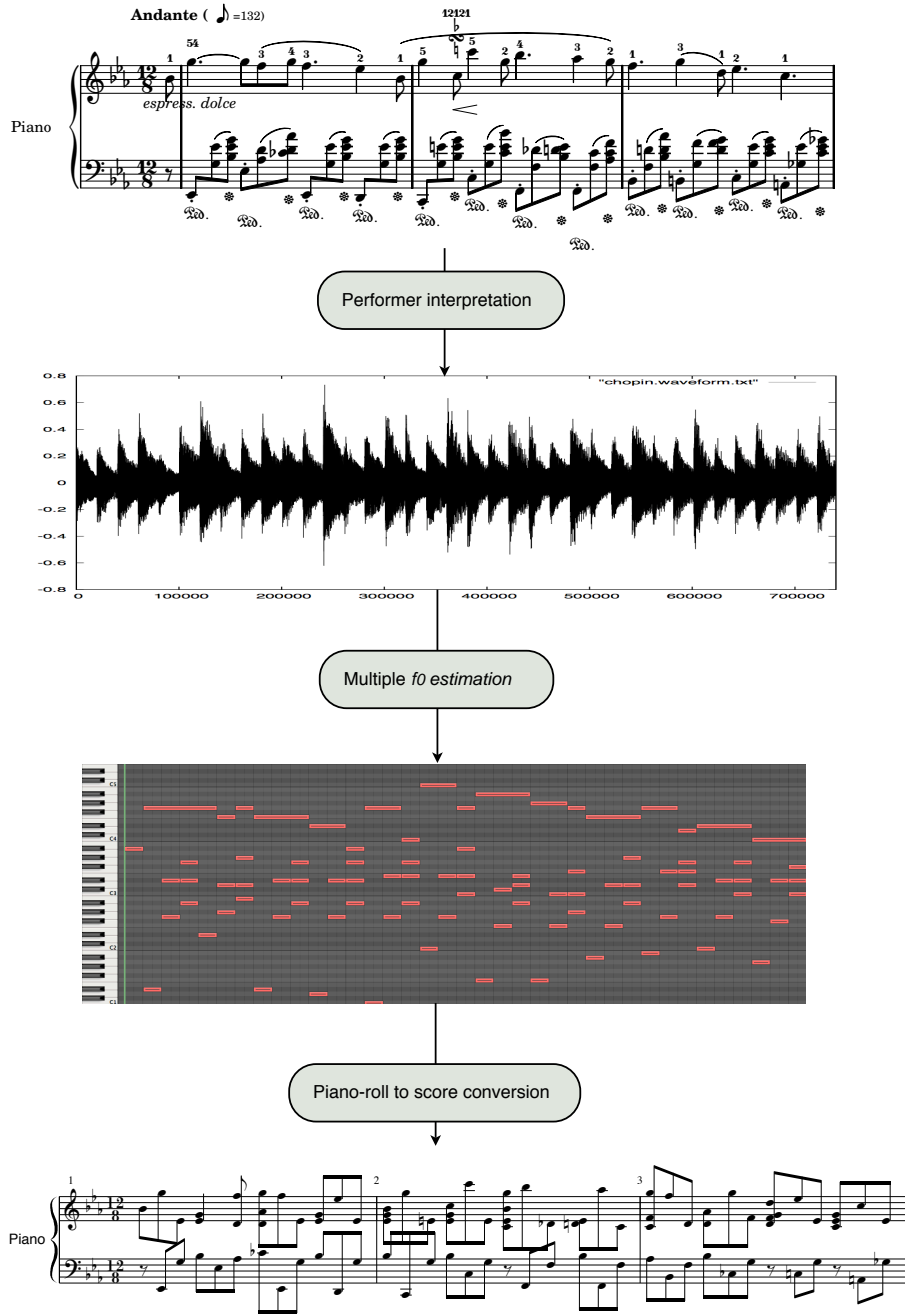


Figure 1.1: Music transcription example from Chopin (Nocturne, Op. 9, N. 2).

1. INTRODUCTION

resulting score would not exactly match the original audio times. This is the reason why the piano-roll is considered as a sound-oriented representation.

This dissertation is mainly focused on the multiple f_0 estimation issue, which is crucial for music transcription. This is an extremely challenging task which has been addressed in several doctoral theses, such as [Moorer \(1975\)](#), [Maher \(1989\)](#), [Marolt \(2002\)](#), [Hainsworth \(2003\)](#), [Cemgil \(2004\)](#), [Bello \(2004\)](#), [Vincent \(2004\)](#), [Klapuri \(2004\)](#), [Zhou \(2006\)](#), [Yeh \(2008\)](#), [Ryynänen \(2008\)](#), and [Emiya \(2008\)](#).

Most multiple f_0 estimation methods are complex and have high computational costs. As discussed in [chapter 3](#), the estimation of multiple simultaneous pitches is a challenging task due to the number of theoretical issues.

The main contributions of this work are a set of novel efficient methods proposed for multiple fundamental frequency estimation ([chapters 6 and 7](#)). The proposed algorithms have been evaluated and compared with other approaches, yielding satisfactory results.

The detection of the beginnings of musical events on audio signals, or onset detection, is also addressed in this work. Onset times can be used for beat tracking, for tempo estimation, and to refine the detection in a multiple f_0 estimation system. A simple and efficient novel methodology for onset detection is described in [chapter 5](#).

The proposed methods have also been applied to other MIR tasks, like genre classification, mood classification, and artist identification. The main idea was to combine audio features with symbolic features extracted from transcribed audio files, and then use a machine learning classification scheme to yield the genre, mood or artist. These combined approaches have been published in ([Lidy et al., 2009, 2007, 2008](#)) and they are beyond the scope of this PhD, which is mainly focused on music transcription itself.

This work is organized as follows. The introductory [chapters 2, 3, and 4](#) describe respectively the theoretical background, the multiple f_0 problem, and the state of the art for automatic music transcription. Then, novel contributions are proposed for onset detection ([5](#)), and multiple fundamental frequency estimation ([6, 7](#)), followed by the overall conclusions and future work ([8](#)).

Outline

- 2 - Background.** This chapter introduces the theoretical background, defining the signal processing, music theory, and machine learning concepts that will be used in the scope of this work.
- 3 - Music transcription.** The multiple f_0 estimation problem and the related theoretical issues are described in this chapter, followed by an introduction to the onset detection problem.
- 4 - State of the art.** This chapter presents an overview of the previous approaches for single f_0 estimation, multiple f_0 estimation, and onset detection. The review is mainly focused on the multiple f_0 estimation methods, proposing a novel categorization of the existing approaches.
- 5 - Onset detection using a harmonic filter bank.** A novel onset detection method based on the properties of harmonic musical sounds is presented, evaluated and compared with other works.
- 6 - Multiple pitch estimation using supervised learning methods.** Novel supervised learning methods are proposed in a simplified scenario, considering synthesized instruments with constant temporal envelopes. For this task, neural networks and nearest neighbors methods have been evaluated and compared.
- 7 - Multiple f_0 estimation using signal processing methods.** Efficient iterative cancellation and joint estimation methods to transcribe real music are proposed in this chapter. These methods have been evaluated and compared with other works.
- 8 - Conclusions and future work.** The conclusions and future work are discussed in this chapter.

2

Background

This chapter describes the signal processing, music theory, and machine learning concepts needed to understand the basis of this work.

Different techniques for the analysis of audio signals based on the Fourier transform are first introduced. The properties of musical sounds are presented, classifying instruments according to their method of sound production and to their spectral characteristics. Music theory concepts are also addressed, describing the harmonic and temporal structures of Western music, and how can it be represented using written and computer notations. Finally, the machine learning techniques used in this work (neural networks and nearest neighbors) are also described.

2.1 Analysis of audio signals

A signal is a physical quantity that is function of one or more independent variables such as time, distance, or pressure. Sounds are air pressure signals which frequencies are in the range that humans can hear (approximately, from 20 to 20,000 Hz¹). The variation of the air pressure amplitude as a function of time within this range is called audio waveform.

A waveform can be modulated into a physical medium as it happens for a magnetic tape. The physical properties of an original sound can also be converted into a sequence of numbers that can be stored in digital form as in a CD.

The accuracy of the conversion of an analog audio waveform $x(t)$ into a digital audio waveform $x[n]$, depends on the sampling rate f_s , which determines how often the sound is sampled. The sampling depth is the maximum numerical size of each sampled value, which is usually expressed as the number of bits involved in coding the samples.

¹In the International System of Units, the unit of frequency is the Hertz. 1 Hz means that an event repeats once per second.

2. BACKGROUND

Systems for automatic music transcription are usually implemented using computers, so the audio signals to be analyzed are in digital form. A digital audio waveform has some drawbacks respect to the analog representation, like aliasing or quantization noise, but these negative effects can be partially reduced using high sampling rates. The most commonly used sampling rate for audio signals is $f_s = 44,100$ Hz, although $f_s = 22,050$ Hz can be sufficient for certain tasks.

2.1.1 Fourier transform

The information of the discrete waveform can be used directly, for example, to detect periodicities in monophonic² sources, by searching for a repetitive pattern in the signal. The waveform also provides information about the temporal envelope, that can be used for some tasks such as beat detection. However, the time domain data is not practical for some approaches that require a different kind of information.

A waveform can be analyzed using the Fourier transform (FT) to map it into the frequency domain. The FT performs the decomposition of a function in a sum of sinusoids of different frequencies, showing the signal within each given frequency band over a range of frequencies. It is a widely used technique for frequency analysis tasks.

The standard Fourier transform (see Eq. 2.1) is well defined for continuous pure sine waves with infinite length.

$$\text{FT}_x(f) = X(f) = \int_{-\infty}^{+\infty} x(t)e^{-j2\pi ft} dt \quad (2.1)$$

The Fourier transform for discrete signals (discrete Fourier transform) is defined in Eq. 2.2. The sequence of complex numbers $x(n)$ is transformed into a sequence of complex numbers $X(k)$. Each spectral bin k is a frequency which value depends on the sampling rate used (see below).

$$\text{DFT}_x(k) = X(k) = \sum_{n=-\infty}^{+\infty} x(n)e^{-j2\pi kn} \quad (2.2)$$

In real world, signals have finite length. $X[k]$ is defined in Eq. 2.3 for a discrete finite signal $x[n]$.

$$\text{DFT}_x[k] = X[k] = \sum_{n=0}^{N-1} x[n]e^{-j\frac{2\pi}{N}kn} \quad , \quad k = 0, \dots, N-1 \quad (2.3)$$

²Only one note playing simultaneously.

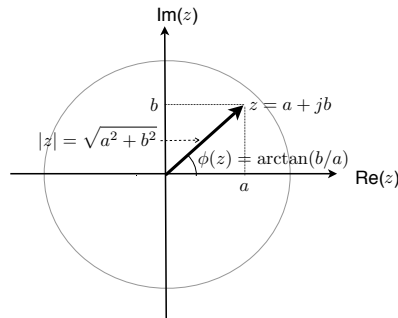


Figure 2.1: Complex plane diagram. Magnitude and phase of the complex number z are shown.

The Shannon theorem limits the number of useful frequencies of the discrete Fourier transform to the Nyquist frequency ($f_s/2$). The frequency of each spectral bin k can be easily computed as $f_k = k(f_s/N)$ since the N bins are equally distributed in the frequency domain of the transformed space. Therefore, the frequency resolution of the DFT is $\Delta f = f_s/N$.

The equations above are described in terms of complex exponentials. The Fourier transform can also be expressed as a combination of sine and cosine functions, equivalent to the complex representation by the Euler's formula.

If the number of samples N is a power of two, then the DFT can be efficiently computed using a fast Fourier transform (FFT) algorithm. Usually, software packages that compute the FFT, like FFTW³ from Frigo and Johnson (2005), use Eq. 2.3, yielding an array of complex numbers.

Using complex exponentials, the radial position or magnitude $|z|$, and the angular position or phase $\phi(z)$ can easily be obtained from the complex value $z = a + jb$ (see Fig. 2.1).

The energy spectral density (ESD) is the squared magnitude of the DFT of a signal $x[n]$. It is often called simply the spectrum of a signal. A spectrum can be represented as a two-dimensional diagram showing the energy of a signal $|X[k]|^2$ as a function of frequency (see Fig. 2.2). In the scope of this work, it will be referred as power spectrum (PS), whereas magnitude spectrum (MS) will be referred represent the DFT magnitudes $|X[k]|$ as a function of frequency.

Spectra are usually plotted with linear amplitude and linear frequency scales, but they can also be represented using a logarithmic scale for amplitude, frequency or both. A logarithmic magnitude widely used to represent the amplitudes is the decibel.

$$\text{dB}(|X[k]|) = 20 \log(|X[k]|) = 10 \log(|X[k]|^2) \quad (2.4)$$

³Fastest Fourier Transform in the West. <http://www.fftw.org>

2. BACKGROUND

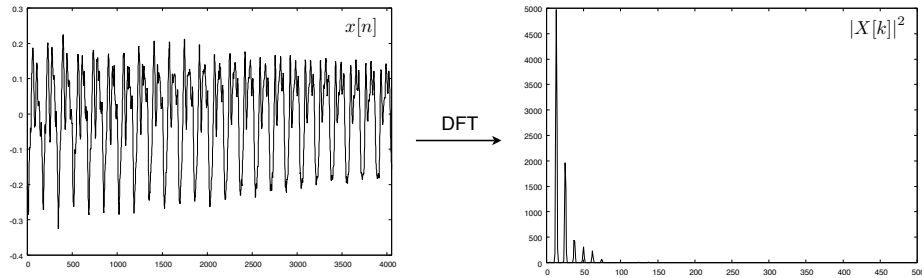


Figure 2.2: Power spectrum of a piano waveform excerpt (I-011PFNOM.wav from [Goto \(2003\)](#) RWC database).

In general, the perceived magnitude of physical variables does not directly correspond with a physical measure. Loudness can be defined as the subjective judgment of the intensity of a sound, and it is the correlate of the physical amplitude. The relation between the physical amplitude and the perceived loudness of a signal is not trivial, and it has been widely studied ([Moore, 1997](#)).

Decibels are related to the human perception of amplitude, but true loudness is a subjective measure which varies from person to person. There exist measures such as A-weighting, based on the work by [Fletcher and Munson \(1933\)](#) to determine loudness level contours for various sound levels, that attempt to get a loudness measure as perceived by an average listener. Sone and Phon are units used to measure the loudness.

Fourier transform limitations

The analysis of discrete and finite signals presents some limitations. First, the continuous to discrete conversion process can produce aliasing and quantization noise. The solution to the aliasing problem is to ensure that the sampling rate is high enough to avoid any spectral overlap or to use an anti-aliasing filter.

The DFT also introduces drawbacks like spectral leakage and the picket fence effect. Spectral leakage is an effect where, due to the finite nature of the analyzed signal, small amounts of energy are observed in frequency components that do not exist in the original waveform, forming a series of lobes in the frequency domain.

The picket fence is an effect related to the discrete nature of the DFT spectrum, which is analogous to looking at it through a sort of picket fence, since we can observe the exact behavior only at discrete points. Therefore, there may be peaks in a DFT spectrum that will be measured too low in level, and valleys that will be measured too high, and the true frequencies where the peaks and valleys are will not be exactly those indicated in the spectrum.

This effect is unavoidable since the computation of the spectrum is limited to integer multiples of the frequency resolution. However, the hidden points can be estimated using interpolation or zero padding. These techniques can not increment frequency resolution, but they allow to estimate the amplitude and frequency of the hidden points.

The direct truncation of the waveform samples (also known as rectangular windowing) leads to undesirable effects like high spectral leakage. To minimize this effect, it is convenient to multiply the samples in the frame by a smooth window shape to remove abrupt edges. Commonly used windows are Hanning, Hamming, Blackman or Blackman-Harris. They have shapes that are generally positive functions, symmetric, and bell shaped. The tradeoffs of using different window functions are compared and discussed by [Harris \(1978\)](#).

Zero padding consists on appending samples with zero values at the end of the input frame in the time domain before doing the DFT. This technique does not increase spectral resolution, but merely interpolates the values of the spectrum at more values.

Usually, in spectral analysis, direct interpolation does not get a better estimation of the hidden points than zero padding. There are different interpolation methods, like linear, quadratic or cubic, with increasing accuracy and cost. Choosing one of these methods depends on the accuracy required, but computational cost must also be considered.

2.1.2 Time-frequency representations

In some applications, it is convenient to represent the spectral information as a function of time. Although there exists only one Fourier transform for a signal, there are several possible time-frequency representations (TFRs).

Short time fourier transform

The most used TFR is the short time Fourier transform (STFT), which represents the Fourier transform of successive signal frames. In the discrete version, the input data is divided into time frames which usually overlap each other. Each input frame is multiplied by a window, then Fourier transformed, and the complex result is added to a matrix, which stores magnitude and phase for each point in time and frequency. Discrete finite STFT can be expressed as:

$$\text{STFT}_x^w[k, m] = \sum_{n=0}^{N-1} x[n]w[n - mI]e^{-j\frac{2\pi}{N}kn} \quad , \quad k = 0, \dots, N - 1 \quad (2.5)$$

where w is the window function, m is the window position index, and I is the hop size.

2. BACKGROUND

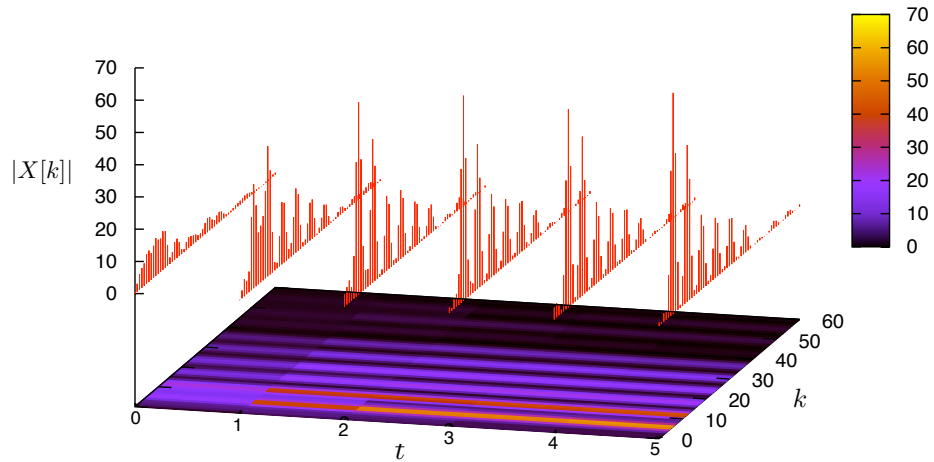


Figure 2.3: Magnitude spectrogram for the beginning section of a piano note. Only the first 60 spectral bins and the first 5 frames are shown. Spectrum at each frame is projected into a plane.

The hop size of the STFT determines how much the analysis starting time advances from frame to frame. Like the frame length (window size), the choice of the hop size depends on the purposes of the analysis. In general, a small hop produces more analysis points and therefore, smoother results across time, but the computational cost is proportionately increased.

Choosing a short frame duration in the STFT leads to a good time resolution and a bad frequency resolution, and a long frame duration results in a good frequency resolution but a bad time resolution. Time and frequency resolutions are conjugate magnitudes, which means that $\Delta f \propto 1/\Delta t$, therefore they can not simultaneously have an arbitrary precision. The decision about the length of the frames in the STFT to get an appropriate balance between temporal and frequency resolution depends on the application.

Spectrograms are three-dimensional diagrams showing the squared magnitude of the STFT evolving in time. Usually, spectrograms are projected into a two-dimensional space (see the lower plane in Fig. 2.3), using colors or grey levels to represent the magnitudes. In the scope of this work, the term magnitude spectrogram will be referred to describe a magnitude spectrum as it changes over time.

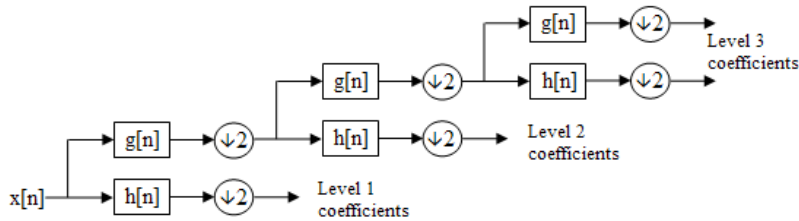


Figure 2.4: Wavelet filter bank with 3 levels.

Discrete wavelet transform

The discrete wavelet transform (DWT) is an alternative to the STFT. It was introduced in order to overcome the limited time-frequency localization of the Fourier transform for non-stationary signals.

Unlike the STFT, the DWT has a variable time-frequency resolution grid with high frequency resolution and low time resolution in the lower frequency area, and high temporal resolution and low frequency resolution on the higher frequency area (see Fig. 2.5(b)). As pointed out by Tzanetakis et al. (2001), the human ear exhibits similar time-frequency resolution characteristics.

The DWT can be generated through an algorithm called multiresolution analysis (MRA), related to sub-band coding, by passing the input signal $x[n]$ through a filter bank (see Fig. 2.4). First, the samples are passed through a low pass filter with impulse response $g[n]$, resulting in a convolution:

$$y[n] = (x * g)[n] = \sum_{k=-\infty}^{\infty} x[k]g[n - k] \quad (2.6)$$

The signal is simultaneously decomposed using a high-pass filter h . The filtered outputs are the wavelet coefficients. It is important that the two filters are related to each other (a quadrature mirror filter must be used). Since half the frequencies of the signal are removed, half the samples can be discarded according to Nyquist theorem, therefore the filter outputs must be subsampled by 2, being $(y \downarrow k)[n] = y[kn]$ the subsampling operation.

A variety of different wavelet transforms have been proposed in the literature, like Haar (1911) or Daubechies (1988) transforms. One of the main drawbacks of the DWT respect to the STFT is the higher computational cost. Extensive reviews of wavelets have been done by Daubechies (1992) and Mallat (1999).

When the mother wavelet is a windowed sinusoid, the wavelet transform can be interpreted as a constant Q Fourier transform.

2. BACKGROUND

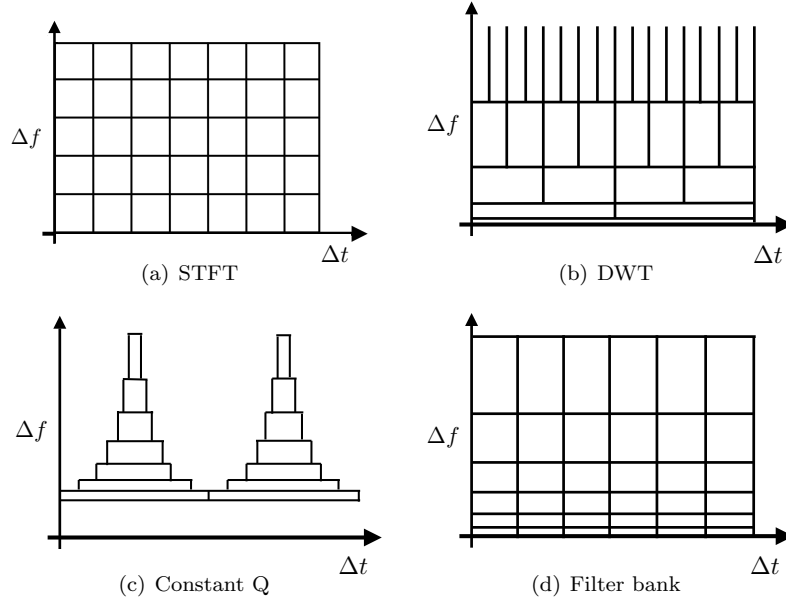


Figure 2.5: Time-frequency resolution grids without overlap for STFT, DWT, constant Q transform from [Brown \(1991\)](#), and a filter bank with 6 bands.

Constant Q transform

Using the Fourier transform, all the spectral bins obtained are equally spaced by a constant ratio $\Delta f = f_s/N$. However, the frequencies of the musical notes (see section 2.3) are geometrically spaced in a logarithmic scale⁴.

The constant Q transform is a calculation similar to the Fourier transform, but with a constant ratio of frequency to resolution Q . This means that each spectral component k is separated by a variable frequency resolution $\Delta f_k = f_k/Q$.

[Brown \(1991\)](#) proposed a constant Q transform in which the center frequencies f_k can be specified as $f_k = (2^{k/b})f_{min}$, where b is the number of filters per octave and f_{min} is the minimum central frequency considered. The transform using $Q = 34$ is similar (although not equivalent) to a 1/24 octave filter bank. The constant Q transform for the k -th spectral component is:

$$X_Q[k] = \frac{1}{N[k]} \sum_{n=0}^{N[k]-1} w[k, n] x[n] e^{-j \frac{2\pi}{N[k]} Q n} \quad (2.7)$$

⁴This scale is also related to the human frequency perception.

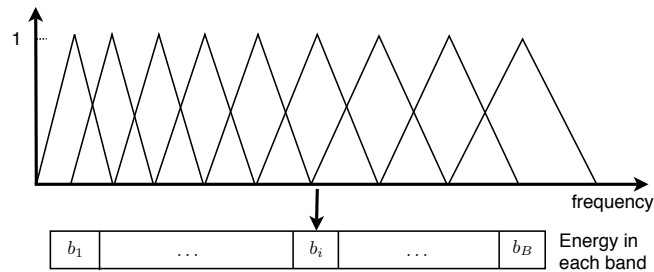


Figure 2.6: Example of a filter bank with triangular shaped bands arranged in a logarithmic frequency scale.

where $N[k]$ is the window size (in samples) used to compute the transform of the frequency k :

$$N[k] = f_s / \Delta f_k = (f_s / f_k) Q \quad (2.8)$$

The window function $w[k, n]$ used to minimize spectral leakage has the same shape but a different length for each component. An efficient implementation of the constant Q transform was described by [Brown and Puckette \(1992\)](#).

The main drawback with this method is that it does not take advantage of the greater time resolution that can be obtained using shorter windows at high frequencies, loosing coverage in the time-frequency plane (see [Fig. 2.5\(c\)](#)).

2.1.3 Filters in the frequency domain

In applications where some frequencies are more important than others, it is useful to isolate certain frequency components. Filters in the frequency domain (see [Fig. 2.5\(d\)](#)) serve to provide information about defined frequency regions, and they can be used to enhance wanted frequency components or to remove unwanted ones.

A filter bank separates the input signal into several components using an array of band pass filters. Each component carries a single frequency subband of the original signal. Therefore, the output of a filter bank is an array of filtered values, each corresponding to the result of filtering the input spectrum through an individual filter (see [Fig. 2.6](#)).

Most filter banks have filters with their edges placed so that they coincide with the center frequencies of adjacent filters. For each filter, frequency components within its pass band are weighted by the magnitude response of the i -th filter $|H_i[k]|$, then squared and summed, as shown in [Eq. 2.9](#).

$$b_i = \sum_{k=0}^{K-1} (|X[k]| \cdot |H_i[k]|)^2 \quad (2.9)$$

Perceptually motivated scales

Psychoacoustic scales have been constructed to imitate the frequency resolution of human hearing. A widely used psychoacoustic scale is the Mel scale introduced by [Stevens et al. \(1937\)](#). A Mel frequency (see [Fig. 2.7](#)) is related to the linear frequency through this relation:

$$\text{Mel}(f) = 2595 \log \left[\frac{f}{700} + 1 \right] \quad (2.10)$$

As [Huang et al. \(2001\)](#) points, one Mel represents one-thousandth of the pitch of 1 kHz, and a doubling of Mels produces a perceptual doubling of pitch.

Other psychoacoustic scale is the Bark introduced by [Zwicker et al. \(1957\)](#), which partitions the hearing bandwidth into perceptually equal frequency bands (critical bands). If the distance between two spectral components is less than the critical bandwidth, then one masks the other.

The Bark scale, also called critical band rate (CBR), is defined so that the critical bands of human hearing have a width of one bark. This partitioning, based on the results of psychoacoustic experiments, simulates the spectral analysis performed by the basilar membrane, in such a way that each point on the basilar membrane can be considered as a bandpass filter having a bandwidth equal to one critical bandwidth, or one bark.

A CBR filter bank is composed of a set of critical band filters, each one corresponding to one bark. The center frequencies to build the filter bank are described by [Zwicker \(1961\)](#). The Mel filter bank is composed of a set of filters with a triangular shape and equally spaced in terms of Mel frequencies. [Shannon and Paliwal \(2003\)](#) showed that Bark and Mel filter banks have similar performance in speech recognition tasks.

The Mel frequency cepstral coefficients (MFCC) have been extensively used in tasks such as automatic speech recognition and music processing. To compute the MFCC features, the power spectrum of the signal is first computed and apportioned through a Mel filter bank. The logarithm of the energy for each filter is calculated before applying a Discrete Cosine Transform (DCT, see [Ahmed et al., 1974](#)) to produce the MFCC feature vector. The DCT of a discrete signal $x[n]$ with a length N is defined as:

$$\text{DCT}_x[i] = \sum_{n=0}^{N-1} x[n] \cos \left[\frac{\pi}{N} i \left(n + \frac{1}{2} \right) \right] \quad (2.11)$$

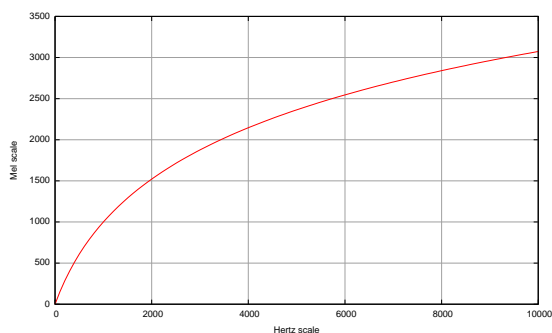


Figure 2.7: Mel scale and linear frequency.

The MFCC are the obtained DCT amplitudes. In most applications, the dimensionality of the MFCC representation is usually reduced by selecting only certain coefficients.

The bandwidth of a filter can be expressed using an equivalent rectangular bandwidth (ERB) measure. The ERB of a filter is defined as the bandwidth of a perfectly rectangular filter with a unity magnitude response and same area as that filter. According to [Moore \(1995\)](#), the ERB bandwidths b_c of the auditory filter at the channel c obey this equation:

$$b_c = 0.108f_c + 24.7 \text{ Hz} \quad (2.12)$$

being f_c the center frequency of the filter.

A filter with a triangular shape can be useful for some applications, but other shapes are needed to model the auditory responses. Filter frequency responses can be expressed in terms of a gaussian function ([Patterson, 1982](#)), a rounded exponential ([Patterson et al., 1982](#)), and a gammatone or “Patterson-Holdsworth” filter ([Patterson et al., 1995](#)). Gammatone filters are frequently used in music analysis, and a description of their design and implementation can be found in ([Slaney, 1993](#)). Auditory filter banks have been used to model the cochlear processing, using a set of gammatone filters uniformly distributed in the critical-band scale.

2.2 Analysis of musical signals

Musical signals are a subset of audio signals, and they have particular characteristics that can be taken into account for their analysis. In this section, some temporal and spectral features of musical sounds are described.

2. BACKGROUND

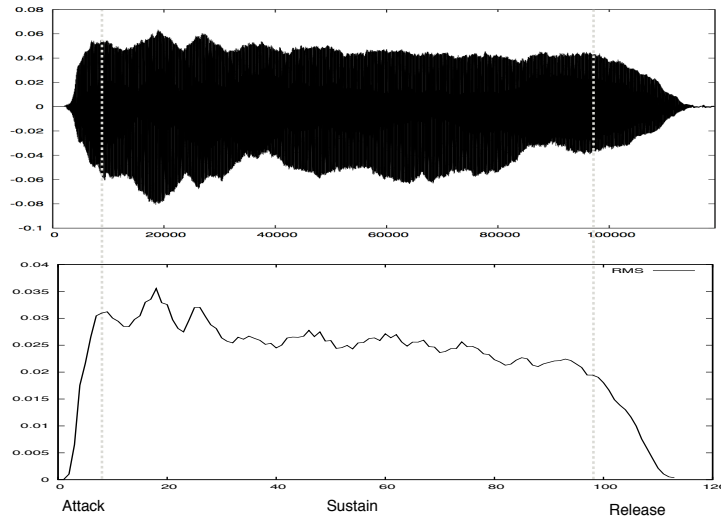


Figure 2.8: Waveform of a flute sound (excerpt of I-331FLNOM from [Goto \(2003\)](#), RWC database) with $f_s = 44,100$ Hz, and RMS levels using $N = 1024$.

2.2.1 Dynamics

Musical instruments produce sounds that evolve in time. The beginning of a sound is known as the onset time, and its temporal end is the offset time. The amplitude envelope refers to a temporally smoothed curve of the sound intensity as a function of time, which evolves from the onset to the offset times.

The envelope of a signal is usually calculated in the time domain by lowpass filtering (with a 30 Hz cut-off frequency) the root mean square (RMS) levels of a signal. The RMS levels $E[n]$ can be obtained as:

$$E[n] = \sqrt{\frac{1}{N} \sum_{i=0}^{N-1} x^2[n+i]} \quad (2.13)$$

where N is the size of the frame. Real sounds have a temporal envelope with an attack and release stages (like percussion or plucked strings), or attack, sustain and decay segments (like woodwind instruments)⁵. The automatic estimation of the intra-note segment boundaries is an open problem, and it has been addressed by some authors like [Jensen \(1999\)](#), [Peeters \(2004\)](#), and [Maestre and Gómez \(2005\)](#).

⁵Synthesizers generate amplitude envelopes using attack, decay, sustain and release (ADSR), but this segmentation is not achievable in real signals, since the decay part is often not clearly present, and some instruments do not have defined sustain or release parts.

The attack of a sound is formally defined as the initial interval during which the amplitude envelope increases. For real sounds, [Peeters \(2004\)](#) considers attack as the initial interval between the 20% and the 80% of the maximum value in the signal, to take into account the possible presence of noise.

Transients are fast varying features characterized by sudden bursts of noise, or fast changes of the local spectral content. During a transient, the signal evolves in a relatively unpredictable way. A transient period is usually present during the initial stage of the sound, and it often corresponds to the period during which the instrument excitation is applied, though in some sounds a transient can also be present in the release stage.

A vibrato is a periodic oscillation of the fundamental frequency, whereas tremolo refers to a periodic oscillation in the signal amplitude. In both cases, this oscillation is of subsonic frequency.

2.2.2 Timbre

In music, timbre is the quality that distinguishes musical instruments. The [American Standards Association \(1960\)](#) defines timbre as that attribute of sensation in terms of which a listener can judge that two sounds having the same loudness and pitch are dissimilar.

From an audio analysis point of view, it is convenient to understand the characteristics that make an instrument different from others. Timbral features extracted from the waveform or the spectrum of an instrument are the basis for automatic instrument classification.

It was shown by [Grey \(1978\)](#) and [Wessel \(1979\)](#) that important timbre characteristics of the orchestral sounds are attack quality (temporal envelope), spectral flux (evolution of the spectral distribution over time), and brightness (spectral centroid).

The spectral flux (SF) is a measure of local spectrum change, defined as:

$$\text{SF}(t) = \sum_{k=0}^{K-1} (\tilde{X}_t[k] - \tilde{X}_{t-1}[k])^2 \quad (2.14)$$

where $\tilde{X}_t[k]$ and $\tilde{X}_{t-1}[k]$ are the energy normalized Fourier spectra in the current and previous frames, respectively:

$$\tilde{X}[k] = \frac{|X[k]|}{\sum_{k=0}^{K-1} |X[k]|} \quad (2.15)$$

The spectral centroid (SC) indicates the position of the sound spectral center of mass, and it is related to the perceptual brightness of the sound. It is calculated as the weighted mean of the frequencies present in the signal, and the weights are their magnitudes.

2. BACKGROUND

$$SC_X = \sum_{k=0}^{K-1} k\tilde{X}[k] \quad (2.16)$$

Besides these features, there exist a number of characteristics that describe a particular timbre. Some of them used for automatic instrument classification are, for instance, spectral skewness, spectral kurtosis, spectral spread, spectral irregularity, spectral roll-off, MFCC, inharmonicity, odd-to-even ratio, tristimulus or temporal centroid. More information about these descriptors can be found in the works from [Peeters \(2004\)](#) and [Herrera et al. \(2006\)](#).

2.2.3 Taxonomy of musical instruments

In the literature, different taxonomies for instrument classification have been proposed. The most referenced is the one proposed by [Hornbostel and Sachs \(1914\)](#), and updated by [Sachs \(1940\)](#), who included the electrophones category. Five families of instruments are considered in this taxonomy:

1. Idiophones, which sound is primarily produced by way of the instrument itself vibrating without the use of membranes or strings. This group includes all percussion instruments apart from drums, and some other instruments. Four subfamilies are considered; struck, plucked, friction and blown idiophones. Cymbals, xylophones, and nail violins belong to this family.
2. Membranophones, which sound is primarily produced by the vibration of a tightly stretched membrane. This group includes all drums and kazoos. Four subfamilies are considered; struck drums, plucked drums, friction drums, and singing membranes. Timpani and snare drums belong to this class.
3. Chordophones, which sound is primarily produced by the vibration of one or more strings. This group includes string instruments and keyboard instruments. Subfamilies are simple and composite chordophones. Pianos, harpsichords, violins, guitars, and harps belong to this family.
4. Aerophones, which sound is primarily produced by vibrating air. This group includes brass and woodwind instruments. Subfamilies are free aerophones and wind and brass instruments. Oboes, saxophones, trumpets, and flutes belong to this category.
5. Electrophones, which sound is produced by electric means. Subfamilies are instruments that have electric action, instruments with electric amplification and radioelectric instruments. Electronic organs, synthesizers and theremins belong to this category.

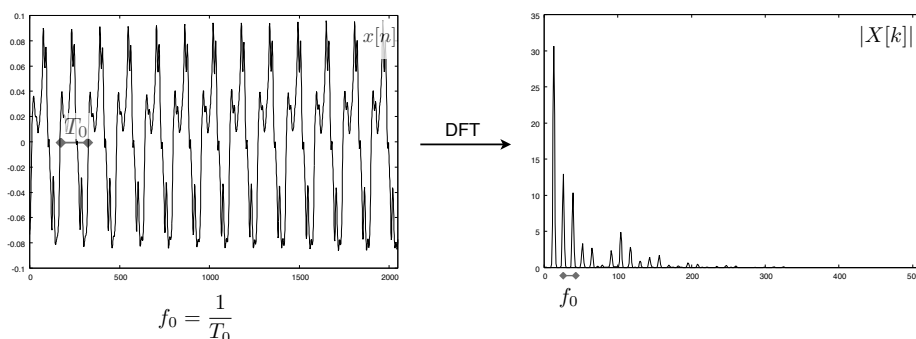


Figure 2.9: Example waveform and spectrum of a violin excerpt (file I-151VNNOM from Goto (2003), RWC database).

Instruments are classified in the families above depending on its exciter, the vibrating element that transforms the energy supplied by the player into sound. However, a complementary taxonomy can be assumed, dividing musical sounds in two main categories: pitched and unpitched sounds.

2.2.4 Pitched musical sounds

The fundamental frequency f_0 of a signal measures the number of occurrences of a repeating event per time unit. Therefore, the fundamental frequency is only present in sounds that are nearly periodic in the time domain. The fundamental period T_0 of a signal is the duration of one cycle in a repeating event, so it is the reciprocal of the fundamental frequency ($T_0 = 1/f_0$).

Pitch is a perceptual attribute related to the fundamental frequency which allows the order of sounds on a frequency related scale extending from low to high (Klapuri, 2006a). More exactly, Hartmann (1996) defines pitch as the frequency of a sine wave that is matched to the target sound by human listeners.

Pitched musical sounds are those that cause a clear pitch sensation, like a piano or a guitar. Most chordophones and aerophones produce pitched sounds, as some electrophones and some idiophones. Pitched musical sounds can be divided into harmonic and inharmonic sounds.

Harmonic sounds

Many Western musical instruments, more precisely those in the chordophone and aerophone families, produce harmonic sounds, as they are based on a harmonic oscillator such as a string or a column of air. The spectrum of these sounds (see Figs. 2.9 and 2.10) shows a series of overtone partials regularly spaced (harmonics).

2. BACKGROUND

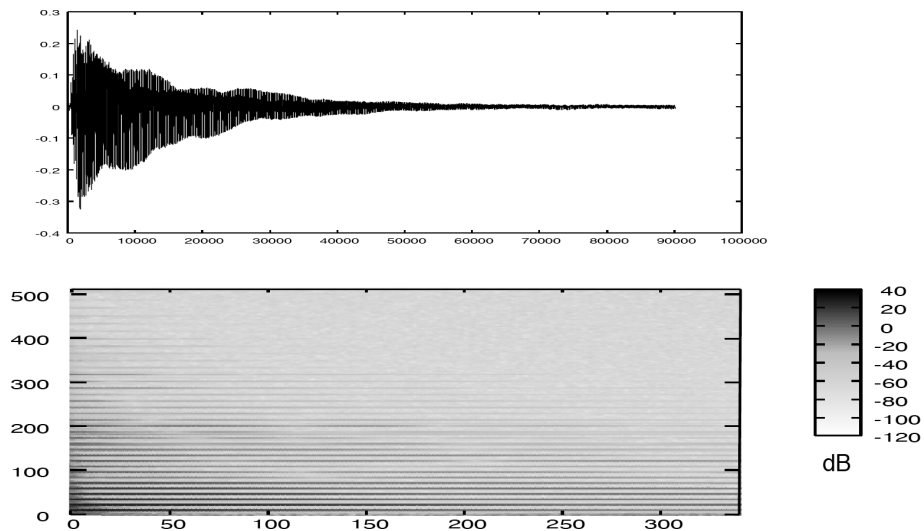


Figure 2.10: Example waveform and spectrogram of a piano note (file I-011PFNOM from [Goto \(2003\)](#), RWC database).

In an ideal harmonic sound, the harmonics are integers multiple of a fundamental frequency. Therefore, the frequency differences of the harmonics corresponds approximately to the fundamental frequency, and the f_0 of a harmonic sound can be defined as the greatest common divisor of the harmonic frequencies.

However, in real harmonic sounds, the partial overtones are usually not exactly multiples of the fundamental partial. This effect is known as inharmonicity, and it appears when the frequency of a partial h is not exactly hf_0 . Although inharmonicity can be a problem for analyzing harmonic sounds, it is not perceptually unpleasant. A slightly inharmonic spectrum adds certain warmth into the sound.

As pointed out by [Schouten \(1940\)](#), the pitch of a complex tone can be perceived even though the frequency component corresponding to the f_0 may not be present (missing fundamental). Note that, if the fundamental were necessary, normal male speech would have no perceivable pitch over the telephone, as frequencies below 300 Hz are generally filtered out⁶.

In most string instruments, the higher order partials gradually shift upwards in frequency. The inharmonicity of a string depends on its physical characteristics, such as tension, stiffness, and length. For instance, a stiff string under low tension (such as a bass string in a piano) exhibits a high degree of inharmonicity,

⁶As pointed out by [Huang et al. \(2001\)](#), the fundamental frequency of speech can vary from 40 Hz for low-pitched male voices to 600 Hz for children or high-pitched female voices.

whereas a thinner string under higher tension (such as a treble string in a piano) or a more flexible string (such as a nylon string used on a guitar or harp) exhibits less inharmonicity.

According to [Fletcher and Rossing \(1988\)](#), the harmonic frequencies in a piano string approximately obey this formula:

$$f_h = hf_0\sqrt{1 + Bh^2} \quad (2.17)$$

A typical value of the inharmonicity factor for the middle pitch range of a piano is $B = 0.0004$, which is sufficient to shift the 17th partial to the ideal frequency of the 18th partial.

In some cases, there are short unpitched excerpts in pitched sounds, mainly in the initial part of the signal. For instance, during the attack stage of wind instruments, the initial breath noise is present before the pitch is perceived. Inharmonic sounds are also produced by the clicking of the keys of a clarinet, the scratching of the bow of violin, or the sound of the hammer of a piano hitting the string, for instance.

The additive synthesis, that was first extensively described by [Moorer \(1977\)](#), is the base of the original harmonic spectrum model, which approximates a harmonic signal by a sum of sinusoids. A harmonic sound can be expressed as a sum of H sinusoids with an error model ϵ :

$$x[n] = \sum_{h=1}^H A_h[n] \cos(2\pi f_h n + \phi_h(0)) + \epsilon[n] \quad (2.18)$$

where A_h is the amplitude of the h -th sinusoid varying in function of time, f_h is the frequency of the sinusoid, and $\phi_h(0)$ is the initial phase.

Most models for parametric analysis of sounds are based on the additive synthesis model developed by [Mcaulay and Quatieri \(1986\)](#) for speech signals, who proposed a robust method for extracting the amplitudes, frequencies, and phases of the component sine waves. The model assumes that the sinusoids are stable partials of the sound with a slowly changing amplitude and frequency. More precisely, instrument harmonics tend to have a linearly increasing angular position and a constant radius.

The model proposed by [Mcaulay and Quatieri \(1986\)](#) was refined by [Serra \(1997\)](#). Spectral modeling synthesis (SMS) is based on modeling sounds as stable sinusoids (harmonics, also called deterministic part), plus noise (residual component, also called stochastic part). Using this model, new sounds can be generated (synthesized) from the analysis of a signal. The analysis procedure detects harmonics by studying the time varying spectral characteristics of the sound and represents them with time varying sinusoids. These partials are then

2. BACKGROUND

subtracted from the original sound and the remaining residual is represented as a time varying filtered white noise component.

Recent parametric models, like the ones proposed by [Verma and Meng \(2000\)](#) and [Masri and Bateman \(1996\)](#), extend the SMS model to consider transients. When sharp onsets occur, the frames prior to an attack transient are similar, and also the frames following its onset, but the central frame spanning both regions is an average of both spectra that can be difficult to be analyzed.

Without considering noise or transients, in a very basic form, a harmonic sound can be described with the relative amplitudes of its harmonics and their evolution over time. This is also known as the harmonic pattern (or spectral pattern). Considering only the spectral magnitude of the harmonics at a given time frame, a spectral pattern \mathbf{p} can be defined as a vector containing the magnitude p_h of each harmonic h :

$$\mathbf{p} = \{p_1, p_2, \dots, p_h, \dots, p_H\} \quad (2.19)$$

This partial to partial amplitude profile is also referred to as the spectrum envelope. Adding the temporal dimension to obtain the spectral evolution in time, a harmonic pattern can be written in matrix notation:

$$\mathbf{P} = \{\mathbf{p}_1, \mathbf{p}_2, \dots, \mathbf{p}_t, \dots, \mathbf{p}_T\} \quad (2.20)$$

In most musical sounds, the first harmonics contain most of the energy of the signal, and sometimes their spectral envelope can be approximated using a smooth curve.

Inharmonic sounds

The pitched inharmonic sounds have a period in the time domain and a pitch, but their overtone partials are not approximately integer multiples of the f_0 . Usually, a vibrating bar is the sound source of these instruments, belonging to the idiophones family. The most common are the marimba, vibraphone (see [Fig. 2.11](#)), xylophone and glockenspiel.

As the analysis of inharmonic pitched sounds is complex and these instruments are less commonly used, most f_0 estimation systems that analyze the signal in the frequency domain do not handle them appropriately.

2.2.5 Unpitched musical sounds

Unpitched musical sounds are those that do not produce a clear pitch sensation. They belong to two main families of the [Hornbostel and Sachs \(1914\)](#) taxonomy: membranophones and idiophones. All the membranophones family, many idiophones and some electrophones produce unpitched sounds.

2.2. ANALYSIS OF MUSICAL SIGNALS

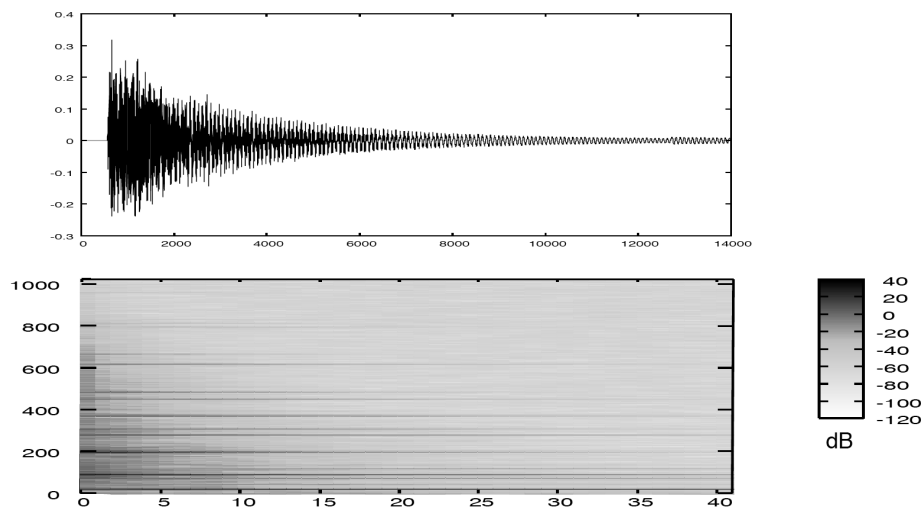


Figure 2.11: Example waveform and spectrogram of a vibraphone (file I-041VIHNM from [Goto \(2003\)](#), RWC database).

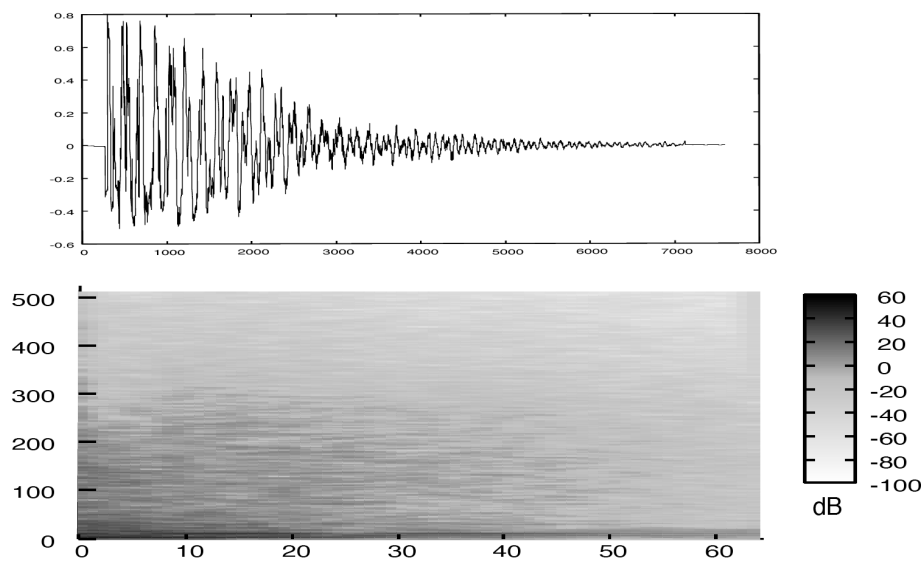


Figure 2.12: Example waveform and spectrogram of an unpitched sound (snare drum, file I-421SD3N3 from [Goto \(2003\)](#), RWC database).

2. BACKGROUND

Most of these sounds are characterized by a sharp attack stage, that usually shows a broad frequency dispersion (see Fig. 2.12). Interestingly, [Fitzgerald and Paulus \(2006\)](#) comment that although synthetic⁷ drum sounds tend to mimic real drums, their spectral characteristics differ considerably from those in real drums.

Spectral centroid, bandwidth of the spectrum and spectral kurtosis are features commonly used in unpitched sound classification.

The transcription of unpitched instruments is referred to the identification of the timbre class and its onset and offset times, as no pitch is present. This task will not be addressed in the scope of this thesis, which is mainly focused on the transcription of pitched sounds. For a review of this topic, see ([FitzGerald, 2004](#)) and ([Fitzgerald and Paulus, 2006](#)).

2.2.6 Singing sounds

According to [Deutsch \(1998\)](#) p.172, singing sounds are produced by the human vocal organ, which consists of three basic components: the respiratory system, the vocal folds and the vocal tract. The respiratory system provides an excess pressure of air in the lungs. The vocal folds chop the airstream from the lungs into a sequence of quasi-periodic air pulses, producing a sound with a fundamental frequency. Finally, the vocal tract modifies the spectral shape and determines the timbre of the voice. The Fig. 2.13 shows a voice spectrogram example.

The term phonation frequency refers to the vibration frequency of the vocal folds and, during singing sounds, this is the fundamental frequency of the generated tone ([Ryynänen, 2006](#)). In a simplified scenario, the amplitudes of the overtone partials can be expected to decrease by about 12 dB per octave ([Sundberg, 1987](#)). The phonation frequencies range from around 100 Hz for male singers over 1 kHz for female singers.

The vocal tract functions as a resonating filter which emphasizes certain frequencies called the formant frequencies. The two lowest formants contribute to the identification of the vowel, and the higher formants to the personal voice timbre.

2.3 Music background

The characteristics of isolated musical audio sounds have been described from a signal processing point of view. However, music is produced by a combination of pitched and/or unpitched overlapped signals beginning at different times. In music, this combination of notes is not random. In most situations, a music

⁷Sounds that are not generated by any real musical instrument.

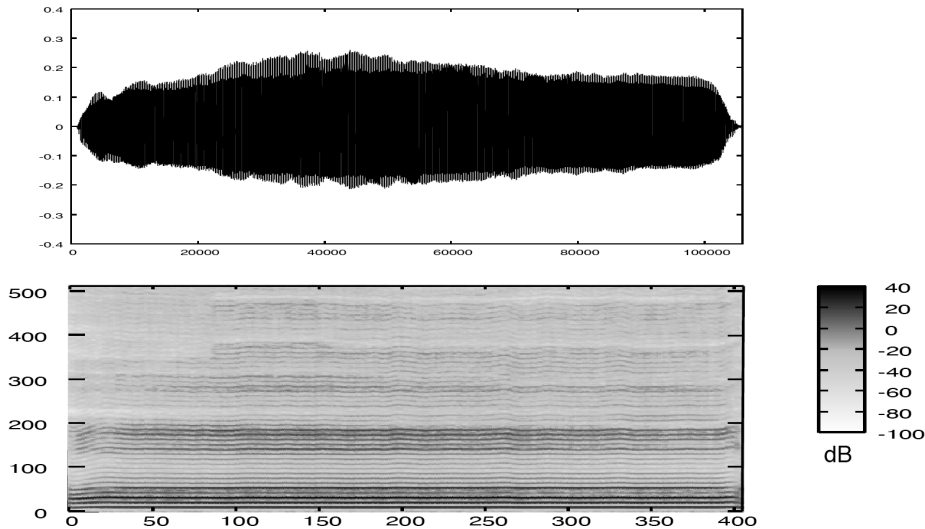


Figure 2.13: Example waveform and spectrogram of a singing male voice, vowel *A* (file I-471TNA1M from [Goto \(2003\)](#), RWC database).

piece follows basic melodic, harmonic and rhythmic rules to be pleasing to most listeners.

In this section, some terms related to the music structure in time and frequency are described, followed by a brief explanation for understanding a musical score and its symbols. Different score formats and representations commonly used in computer music are also introduced.

2.3.1 Tonal structure

Harmony is a term which denotes the formation and relationships of simultaneous notes, called chords, and over time, chordal progressions. A melody is a sequence of pitched sounds with musically meaningful pitches and a metrical structure. Therefore, the term melody refers to a sequence of pitches, whereas harmony refers to the combination of simultaneous pitches.

A musical interval can be defined as a ratio between two pitches. The term harmonic interval refers to the pitch relationship between simultaneous notes, whereas melodic interval refers to the pitch interval of two consecutive notes. In Western tonal music, intervals that cause two notes to share harmonic positions in the spectrum, or consonant intervals, are more frequent than those without harmonic relationships, or dissonant intervals.

2. BACKGROUND

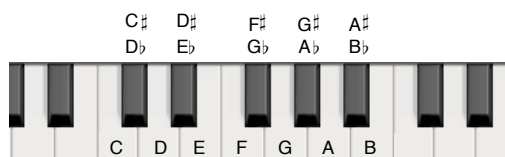


Figure 2.14: Western note names in a piano keyboard. Only one octave is labeled.

Musical temperaments

In terms of frequency, musical intervals are relations described by the ratio between the respective frequencies of the involved notes. The octave is the simplest interval in music, after the unison⁸. Two notes separated by one octave have a frequency ratio of 2:1. The human ear tends to hear two notes an octave apart as being essentially the same. This is the reason why, in most musical cultures (like Western, Arabic, Chinese, and Indian music), the wide range of pitches is arranged across octaves in a logarithmic frequency scale.

Music is based on the octave relationship, but there exist different ways for arranging a number of musical notes within an octave and assigning them a given frequency. In Western music, the most common tuning system is the twelve tone equal temperament, which divides each octave into 12 logarithmically equal parts, or semitones.

In this musical temperament, each semitone is equal to one twelfth of an octave. Therefore, every pair of adjacent notes has an identical frequency ratio of $1:2^{1/12}$, or 100 cents. One tone is defined as a two semitones interval. Equal temperament is usually tuned relative to a standard frequency for pitch A of 440 Hz⁹.

Western musical pitches

A musical note can be identified using a letter (see Fig. 2.14), and an octave number. For instance, C_3 refers to the note C from the third octave. Notes separated by an octave are given the same note name. The twelve notes in each octave are called pitch classes. For example, the note C_3 belongs to the same pitch class than C_4 .

⁸An unison is an interval with a frequency ratio 1:1.

⁹This frequency reference for tuning instruments was first adopted as the USA Standard Pitch in 1925, and it was set as the modern concert pitch in May 1939. Before, a variety of standard frequencies were used. For example, in the time of Mozart, the pitch A had a value close to 422 Hz.

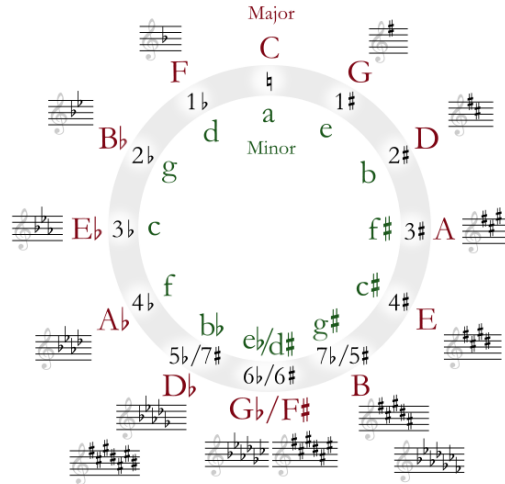


Figure 2.15: Musical major keys (uppercase), and minor keys (lowercase). The number of alterations and the staff representation are shown. Fig. from http://en.wikipedia.org/wiki/File:Circle_of_fifths_deluxe_4.svg.

There are 12 pitch classes, but only 7 note names (C,D,E,F,G,A,B). Each note name is separated by one tone except F from E, and C from B, which have a one semitone interval. This is because modern music theory is based on the diatonic scale.

Musical scales and tonality

The diatonic scale is a seven note musical scale comprising five whole steps and two half steps, with the pattern repeating at the octave. The major scale is a diatonic scale which pattern of intervals in semitones is 2-2-1-2-2-2-1, starting from a root note¹⁰. For instance, the major diatonic scale with root note C is built using the white keys of the piano. The natural minor scale has a pattern of intervals 2-1-2-2-1-2-2.

In tonal music, a scale is an ordered set of notes typically used in a tonality (also referred as key). The tonality is the harmonic center of gravity of a musical excerpt. Intervals in the major and minor scales are consonant intervals relative to the tonic, therefore they are more frequent within a given key context.

¹⁰The root note is also called tonic or harmonic center

2. BACKGROUND

Harmonic Series

Harmonic	Frequency	Nearest Tone	Interval Formed
9	2358	D ₇	Major Second M2, 9:8
8	2096	C ₇	
7	1834	-Bb ₆	Minor Sixth m6, 8:5
6	1572	G ₆	
5	1310	E ₆	Minor Third m3, 6:5
4	1048	C ₆	Major Third M3, 5:4
3	786	G ₅	Perfect Fourth P4, 4:3
2	524	C ₅	Perfect Fifth P5, 3:2
1	262	C ₄	Octave 2:1

Figure 2.16: Harmonics and intervals. The first nine harmonics of middle C. Their frequencies and nearest pitches are indicated, as well as the Western tonal-harmonic music intervals. Fig. from [Krumhansl \(2004\)](#).

A musical excerpt can be arranged in a major or a minor key (see Fig. 2.15). Major and minor keys which share the same signature are called relative. Therefore, C major is the relative major of A minor, whereas C minor is the relative minor of E major. The key is established by particular chord progressions.

Consonant and dissonant intervals

As introduced above, harmonic and melodic intervals can be divided into consonant and dissonant. Consonant intervals are those that cause harmonic overlapping in some degree. For harmonic interference it is not required exact frequency overlap, only approximation¹¹.

The perceptual dimension of consonance and dissonance is related to ratios of frequencies. The ordering along the dimension of consonance corresponds quite closely to the size of the integers in the ratios ([Vos and Vianen, 1984](#)). The unison (1:1) and octave (2:1) are the most consonant intervals, followed by the perfect intervals. Perfect intervals¹² are the perfect fifth (3:2) and the perfect fourth (4:3). The major third (5:4), minor third (6:5), major sixth (5:3),

¹¹Other temperaments, like the meantone temperament, make the intervals closer to their ideal just ratios.

¹²In the equal temperament, besides the unison and the octave, the interval ratios described are approximate.

and minor sixth (8:5) are next most consonant. The least consonant intervals in western harmony are the minor second (16:15), the major seventh (15:8) and the tritone (45:32).

In music, consonant intervals are more frequent than dissonant intervals. According to [Kosuke et al. \(2003\)](#), trained musicians find more difficult to identify pitches of dissonant intervals than those of consonant intervals.

It is hard to separate melody from harmony in practice ([Krumhansl, 2004](#)), but harmonic and melodic intervals are not equivalent. For example, two notes separated by one octave play the same harmonic rule, although they are not interchangeable in a melodic line.

The most elemental chord in harmony is the triad, which is a three note chord with a root, a third degree (major or minor third above the root), and a fifth degree (major or minor third above the third).

2.3.2 Rhythm

A coherent temporal structure is pleasing to most listeners. Music has a rhythmic dimension, related to the placement of sounds at given instants and their accents¹³.

In the literature, there exist some discrepancies about the terminology used to describe rhythm¹⁴. An excellent study of the semantics of the terms used in computational rhythm can be found in ([Gouyon, 2008](#)).

According to [Fraisse \(1998\)](#), a precise, generally accepted definition of rhythm does not exist. However, as [Honing \(2001\)](#) points out, there seems to be agreement that the metrical structure, the tempo (tactus) and the timing are three main rhythmic concepts.

The metrical structure refers to the hierarchical temporal structure, the tempo indicates how fast or slow is a musical piece, and the timing deviations that occur in expressive performances are related to the temporal discrepancies around the metrical grid.

Metrical structure

Meter is a concept related to an underlying division of time. From a perceptual point of view, [Klapuri \(2003b\)](#) considers musical meter as a hierarchical structure consisting of pulse sensations at different levels (time scales). Usually, three main levels are considered in a metrical structure; the beat (tactus), the measure (bar) and the tatum.

The beat, or tactus level, is the basic time unit in music. [Handel \(1989\)](#) defines beat as a sense of equally spaced temporal units. It typically corresponds

¹³In music, an accent is an emphasis placed on a particular note.

¹⁴For example, in some works, the terms pulse and beat are equivalent.

2. BACKGROUND

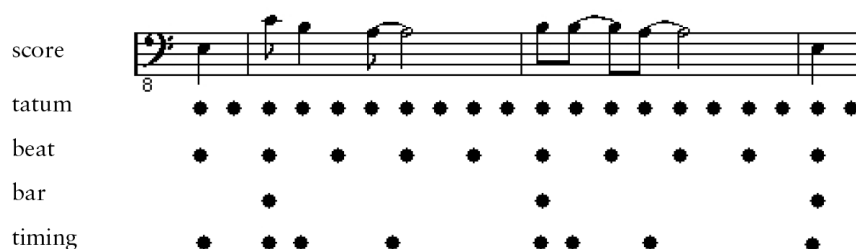


Figure 2.17: Diagram of relationships between metrical levels and timing. Fig. from [Hainsworth \(2003\)](#).

to the preferred human foot tapping rate ([Klapuri et al., 2006](#)), or to the dance movements when listening to a musical piece.

A measure constitutes a temporal pattern and it is composed by a number of beats. In Western music, rhythms are usually arranged with respect to a time signature. The time signature (also known as meter signature) specifies how many beats are in each measure and what note value constitutes one beat. One beat usually corresponds to the duration of a quarter note¹⁵ (or *crochet*) or an eighth note (or *quaver*) in musical notation. A measure is usually 2, 3, or 4 beats long (*duple*, *triple*, or *quadruple*), and each beat is normally divided into 2 or 3 basic subdivisions (*simple*, or *compound*). Bar division is closely related to harmonic progressions.

Unfortunately, the perceived beat does not always correspond with the one written in a time signature. According to [Hainsworth \(2003\)](#), in fast jazz music, the beat is often felt as half note (or *minim*), i.e., double of his written rate, whereas hymns are often notated with the beat given in minims, the double of the perceived rate.

The *tatum*¹⁶, first defined by [Bilmes \(1993\)](#), is the lowest level of the metric musical hierarchy. It is a high frequency pulse that we keep in mind when perceiving or performing music. An intuitive definition of the *tatum* proposed by [Klapuri \(2003b\)](#) refers it as the shortest durational value in music that are still more than incidentally encountered, i.e., the shortest commonly occurring time interval. It frequently corresponds to a binary, ternary, or quaternary subdivision of the musical beat. The duration values of the other notes, with few exceptions, are integer multiples of the *tatum*. The *tatum* is not written in a modern musical score, but it is a perceptual component of the metrical structure.

¹⁵Note durations are shown in Fig. 2.1.

¹⁶In honor of Art Tatum.

The generative theory of tonal music (GTTM), introduced by [Lerdahl and Jackendoff \(1983\)](#), is a detailed theory of musical hierarchies from a psychological perspective, attempting to produce formal descriptions in a scientific sense. In the GTTM, besides metrical structure, grouping structure is also considered. The grouping structure is related to the musical surface, where listeners can find motives, phrases and sections.

Tempo

The tempo (also referred as *tactus*) indicates the speed of the underlying beat. It is usually measured in bpm (number of beats per minute), and it is inversely proportional to the beat period. Having a beat period T_b expressed in seconds, the tempo can be computed as $\mathcal{T} = 60/T_b$. Like other rhythmic components, it can vary along a piece of music.

Timing

Usually, when a score is performed by a musician, the onset times of the played notes do not exactly correspond with those indicated in the score. This temporal deviation, known as timing deviation (see [Fig. 2.17](#)) is frequent in musical signals. It can be produced either by slight involuntary deviations in the performance, or by deliberate expressive rhythm alterations, like swing.

In music psychology, emotion component of music is strongly associated with music expressivity ([Juslin et al., 2006](#)). A musical piece can be performed to produce different emotions in the listener (it can be passionate, sweet, aggressive, humorous, etc). In the literature, there exist a variety of approaches to map a song into a psychologically based emotion space, classifying it according to its mood. Timing and metrical accents are frequently affected by mood alterations in the performance.

2.3.3 Modern music notation

A score is a guide to perform a piece of music. In the history of music, a number of different representations of music through written symbols have been used. The most extended musical notation is the modern notation, originated in European classical music to represent Western tonal music.

Modern notation is only intended for the representation of pitched sounds, and it represents the music using a two dimensional space. A five line staff is used as basis for this notation, where pitches are represented in the vertical axis, and time in the horizontal axis.

Pitch is shown by placement of notes on the staff, over a line or between two lines. The pitch can be modified by accidentals, like sharps (\sharp), flats (b), double sharps (\times), double flats (bb), or naturals (\natural). Sharps and flats increase

2. BACKGROUND















Note	Rest	American name	British name
		Whole	Semibreve
		Half	Minim
		Quarter	Crotchet
		Eighth	Quaver
		Sixteenth	Semiquaver
		Thirty-second	Demisemiquaver
		Sixty-fourth	Hemidemisemiquaver

Table 2.1: Symbols used to represent the most frequent note and rest durations.

or decrease the pitch by one semitone, respectively. Notes with a pitch outside of the range of the five line staff can be represented using ledger lines, which provide a single note with additional lines and spaces.

Duration is shown with different note figures (see Fig. 2.1), and additional symbols such as dots (\cdot) and ties (\smile). Notation is read from left to right.

A staff begins with a clef, which indicates the pitch of the written notes. Following the clef, the key signature indicates the key by specifying certain notes to be flat or sharp throughout the piece, unless otherwise indicated.

The time signature appears after the key signature. Measures (bars) divide the piece into regular groupings of beats, and the time signatures specify those groupings.

Directions to the performer regarding tempo and dynamics are added above or below the staff. In written notation, the term dynamics usually refers to the intensity of the notes¹⁷. The two basic dynamic indications in music are **p** or piano, meaning soft, and **f** or forte, meaning loud or strong.

In modern notation, lyrics can be written for vocal music. Besides this notation, others can be used to represent unpitched instruments (percussion notation) or chord progressions (e.g., tablatures).

2.3.4 Computer music notation

A digital score contains symbolic data which allow the easy calculation of musical information and its manipulation. In a computer, a score can be stored

¹⁷Although the term dynamics is sometimes used to refer other aspects of the execution of a given piece, like staccato, legato, etc.

Study in C# minor

S. Heller (1813-1885)
op.81 no. 10

Allegro leggiero
(♩ = c.120-138)

Key signature

Tempo

Clefs

Time signature

Dynamic indications

Figure 2.18: Excerpt of a musical score in modern notation. Symbols that do not represent notes are annotated.

and represented in different ways. Musical software can decode symbolic data and represent them in modern notation. Software like sequencers can also play musical pieces in symbolic formats using a synthesizer.

Symbolic formats

The MIDI¹⁸ (Musical Instrument Digital Interface) protocol, introduced by the MIDI Manufacturers Association and the Japan MIDI Standards Committee in 1983, enables electronic musical instruments, computers, and other equipment to communicate, control, and synchronize with each other. MIDI does not generate any sound, but it can be used to control a MIDI instrument that will produce the specified sound. Event messages such as the pitch and intensity (velocity) of musical notes can be transmitted using this protocol. It can also control parameters such as volume, vibrato, panning, musical key, and clock signals to set the tempo.

In MIDI, the pitch of a note is encoded using a number (see Fig. 2.19). A frequency f can be converted into a MIDI pitch number n using Eq. 2.21:

$$n = \text{round} \left(69 + 12 \log_2 \left[\frac{f}{440} \right] \right) \quad (2.21)$$

Inversely, the frequency f of a given MIDI pitch number n can be obtained using Eq. 2.22.

$$f = 440 \cdot 2^{\frac{n-69}{12}} \quad (2.22)$$

Other important MIDI component is the intensity of a note, which is encoded using a number in the range $[0 : 127]$.

¹⁸<http://www.midi.org>

2. BACKGROUND

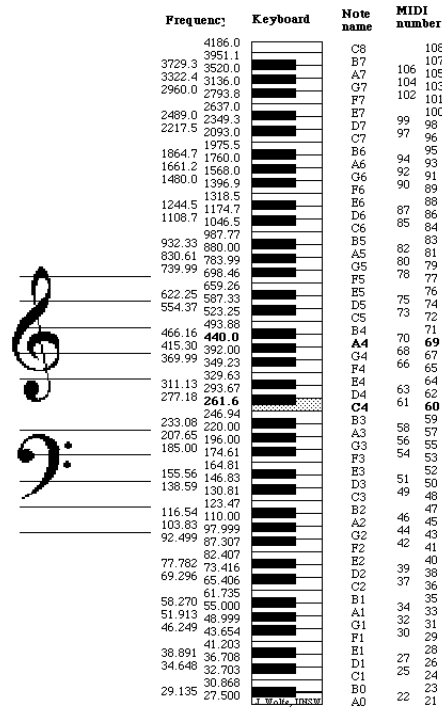


Figure 2.19: Equal temperament system, showing their position in the staff, frequency, note name and MIDI note number. Fig. from Joe Wolfe, University of South Wales (UNSW), <http://www.phys.unsw.edu.au/jw/notes.html>.

MIDI messages, along with timing information, can be collected and stored in a standard MIDI file (SMF). This is the most extended symbolic file format in computer music. The SMF specification was developed by the MIDI Manufacturers Association (MMA). Large collections of files in this format can be found on the web.

The main limitation of MIDI is that there exist musical symbols in modern notation that can not be explicitly encoded using this format. For example, pitch names have a different meaning in music, but there is no difference between $C\sharp$ and $D\flat$ in MIDI, as they share the same pitch number. In the literature, there are a number of pitch spelling algorithms to assign contextually consistent letter names to pitch numbers according to the local key context. A comparison of these methods can be found in the [Meredith and Wiggins \(2005\)](#) review.

This is because MIDI is a sound-oriented code. It was designed as a protocol to control electronic instruments that produce sound, and not initially intended to represent musical scores. Another example of a sound related code is CSound score format, which was also designed for the control and generation of sounds.

CSound is a software synthesizer developed by the Medialab at MIT. For specification details of the language, see (Vercoe, 1991), and (Boulangier, 1999).

There exist other schemes, referred as musical notation codes, that overcome the limitations of the MIDI specification for score representation. Some examples are MusicXML¹⁹, SCORE²⁰, and Lilypond²¹.

A third category of musical codes are those designed for musical data management and analysis, like Humdrum²², and the Essen Associative Code²³.

The work from Selfbridge-Field (1997) presents an extensive review for a variety of musical coding schemes, including MIDI.

Visual representations of SMFs

The symbolic musical information stored in a SMF can be represented in different ways by a computer software, as it is not a symbolic code well defined to obtain a modern score. The musical data of a SMF can be visualized as a piano-roll or as approximate modern notation. Different sequencers provide very similar piano-roll representations of a MIDI file, but they generate different scores from the same SMF, as MIDI data must be interpreted to be translated into a score.

The piano-roll is a natural representation of a SMF with time in the horizontal axis, and pitch in the vertical axis (see Fig. 2.20, top). Notes are represented using horizontal bars in the time-frequency grid. Some sequencers assign a color to the bars related to the note intensity.

Translating MIDI data into modern notation leads to quantization. Quantization is the process of aligning the musical notes to a given grid. Note onsets²⁴ are set on their nearest beats or exact fractions of beats, depending on the quantization unit. This process is usually done for representing a readable score removing slight timing deviations.

Using quantization, some musical figures like triplets²⁵ or notes shorter than the quantization unit (like those in bar 10, Fig. 2.20) can not be represented correctly.

More information about the problem of extracting the musical score from a MIDI performance can be found in (Cambouropoulos, 2000). Although the MIDI format has some limitations, as pointed out by Klapuri and Davy (2006), standard MIDI files are usually considered a valid format for representing symbolic music in computational music transcription.

¹⁹<http://www.recordare.com/xml.html>

²⁰<http://www.scoremus.com/score.html>

²¹<http://lilypond.org>

²²<http://www.humdrum.org/Humdrum/>

²³<http://www.esac-data.org/>

²⁴Quantization can be done only for onsets, or for onsets and offsets.

²⁵Three triplet quarter notes durations sum a half note duration, so the duration of a triplet quarter note is 2/3 the duration of a standard quarter note.

2. BACKGROUND

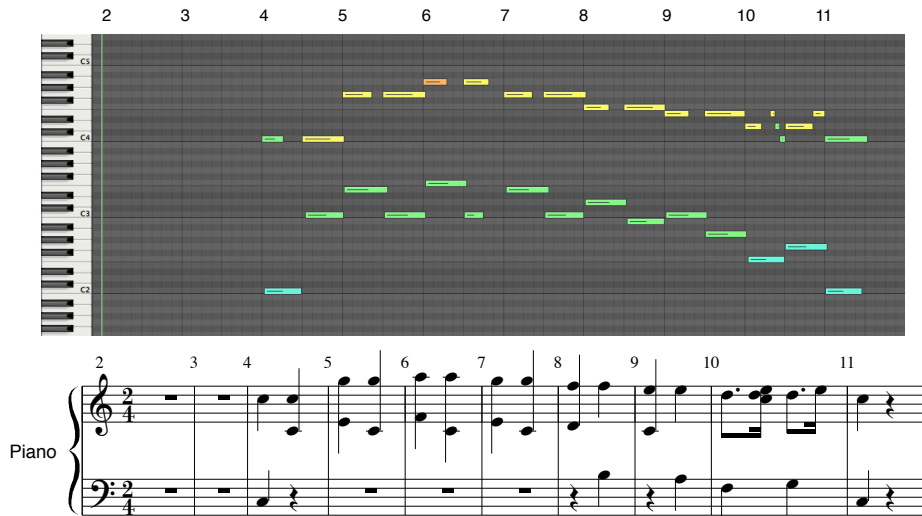


Figure 2.20: Example of a piano-roll (top) and score representation (bottom) for an excerpt of the MIDI file RWC-MDB-C-2001-27 from [Goto \(2003\)](#), RWC database, W.A. Mozart variations on ‘Ah Vous Dirai-je Maman’, K.265/300e. Figs. obtained using Logic Pro 8.

2.4 Supervised learning

Supervised learning methods attempt to deduce a function from a training set. The training data consist of pairs of input data (usually, vectors), and desired outputs. After the learning stage, a supervised learning algorithm can predict the value of the function for any valid input data. The basic concepts to understand the supervised learning methods used in Chapter 6 for multiple f_0 estimation are briefly described next.

2.4.1 Neural networks

The multilayer perceptron (MLP) or multilayer neural network architecture was first introduced by [Minsky and Papert \(1969\)](#). Citing [Duda et al. \(2000\)](#), multilayer neural networks implement linear discriminants, but in a space where the inputs have been nonlinearly mapped. The key power provided by such networks is that they admit fairly simple algorithms where the form of the nonlinearity can be learned from training data.

The Fig. 2.21 shows an example of a MLP. This sample neural network is composed by three layers: the input layer (3 neurons), a hidden layer (2 neurons) and output layer (3 neurons), connected by weighted edges.

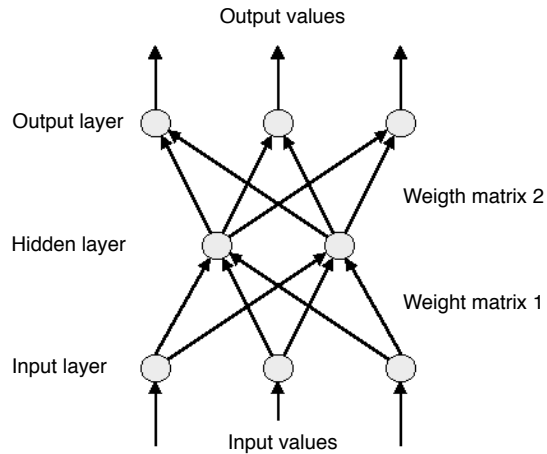


Figure 2.21: Multilayer perceptron architecture.

The weights of a multilayer neural network can be learned by the standard backpropagation algorithm from [Rumelhart et al. \(1986\)](#). Each feature of the input vector is presented to one neuron of the input layer, whereas each hidden unit performs the weighted sum of its inputs to form its activation, and each output unit similarly computes its activation based on the hidden unit signals ([Duda et al., 2000](#)).

A transfer function²⁶, which is typically a sigmoid, is used to determine whether a neuron is activated or not according to its input.

Time-delay neural networks

As pointed out by [Hush and Horne \(1993\)](#), time-delay neural networks ([Waibel, 1989](#)) are considered as non-recurrent dynamic networks, although essentially are like static nets traversing temporal series. This kind of nets can model systems where the output $y[t]$ has a dependence of a limited time interval in the input $u[t]$:

$$y(t) = f[u(t-m), \dots, u(t), \dots, u(t+n)] \quad (2.23)$$

Using this network, time series can be processed as a collection of static input-output patterns, related in short-term as a function of the width of the input window. The TDNN architecture is very similar to a MLP, but the main difference is that the input layer is also fed with information about adjacent time frames. Each hidden unit accepts input from a restricted (spatial) range

²⁶Also called activation function.

2. BACKGROUND

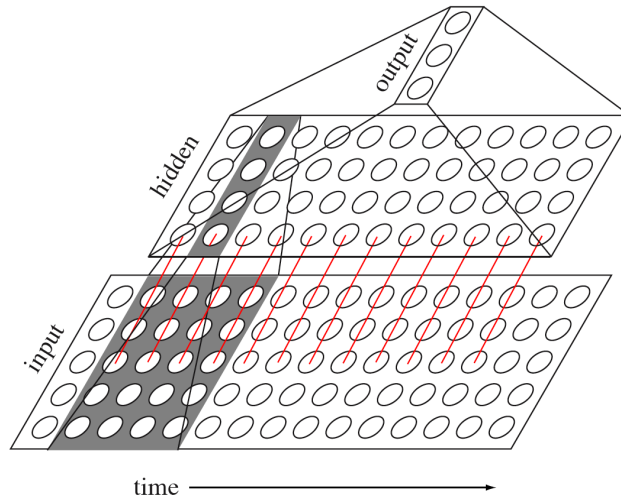


Figure 2.22: TDNN architecture. Fig. from [Duda et al. \(2000\)](#).

of positions in the input layer (see Fig. 2.22). A TDNN can be trained with the same standard backpropagation algorithm used for a MLP.

2.4.2 Nearest neighbors

The nearest neighbors (NN) algorithm is a non-parametric method proposed by [Cover and Hart \(1967\)](#) to classify objects based on their closest training examples in a feature space. Given an unknown sample, the NN algorithm finds in the training set the closest prototype in a n -dimensional feature space, classifying the test sample in the same class than the prototype.

Unlike neural networks, this is a type of instance-based learning where the function is only approximated locally, therefore no new classes (or prototypes) can be directly derived except from the ones that are present in the training stage. For this reason, the k NN algorithm is sensitive to the local structure of the data.

The success rates obtained with NN can be improved using a number of k nearest neighbors. A k nearest neighbor (k NN) algorithm finds the k nearest neighbors of a sample, assigning it to the class with most representatives through a voting process.

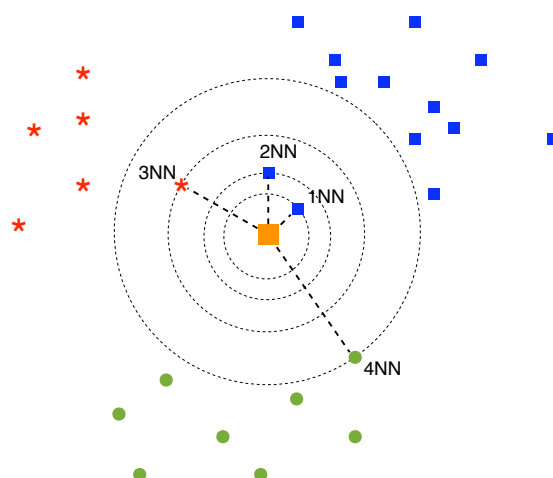


Figure 2.23: Simplified example of NN classification in a two-dimensional feature space. Using the Euclidean distance, the test sample is classified as the blue squared class.

Usually, the Euclidean distance is chosen to measure the proximity in the feature space, though other metrics, such as Manhattan and Minkowsky distances²⁷, can be used instead.

²⁷More information about different metrics used for NN classification can be found in (Duda et al., 2000), section 4.6.

3

Music transcription

This chapter briefly addresses the human music transcription process, followed by an analysis of the theoretical issues in automatic transcription from a signal processing point of view. Finally, the onset detection task is also introduced. The different metrics used for the evaluation of multiple f_0 estimation and onset detection methods are also discussed.

3.1 Human music transcription

Trained listeners can usually get better transcriptions than automatic systems. This can be justified since the music transcription problem is similar to speech recognition, where human training and experience play an important role. Most automatic transcription systems consider isolated frames (sometimes, few adjacent frames), whereas humans consider a wider context. Developing algorithms that consider a wide context is a challenging task due to the high computational cost required.

Some studies have been done to analyze the processes that trained musicians use for transcription. It is necessary to have musical skills to write down the listened notes into a musical notation. This ability is acquired through musical dictation practice. However, it is not necessary much training for recognizing, in some way, the chords, the melody and the rhythm in a musical piece, and we can easily remember and sing a musical piece. Therefore, musical skills can be split into written skills, related to notation abilities, and aural skills, related to the perceptual understanding of music.

Memory is very important in the transcription process. Citing [Sloboda \(1985\)](#), “the way one hears music is crucially dependent upon what one can remember of past events in the music ... To perceive an event musically (that is, to recognize at least part of its musical function) is to relate it to past events”.

We can hear key modulations if we can remember the previous key, and a note or chord has no musical significance without considering the preceding or

3. MUSIC TRANSCRIPTION

following events. We can identify relative pitch differences, more than absolute pitches. Another proof of the importance of the musical context is that pitches can be very hard to identify in a confusing context like, for instance, when two different songs are heard simultaneously.

[Klapuri et al. \(2000\)](#) performed a test to measure the pitch identification ability of trained musicians using isolated chords. The results were compared with those obtained using an automatic transcription system, and only the two most skilled subjects performed better than the computational approach, showing that it is not easy to analyze notes out of context.

[Hainsworth \(2003\)](#) proposed a test where trained musicians were asked to answer how did they perform transcription. A common pattern was found. The first step was to do a rough structural analysis of the piece, breaking the song into sections, finding repetitions, and in some cases marking key phrases. Then, a chord scheme or the bass line were detected, followed by the melody. Finally, the inner harmonies were heard by repeated listening, building up a mental representation. According to [Hainsworth \(2003\)](#), “no-one transcribes anything but simple music in a single pass”.

The auditory scene analysis (ASA) is a term proposed by [Bregman \(1990\)](#) to describe the process by which the human auditory system organizes sound into perceptually meaningful elements. In computational analysis, the related concept is called computational auditory scene analysis (CASA), which is closely related to source separation and blind signal separation. The key aspects of the ASA model are segmentation, integration, and segregation. The grouping principles of ASA can be categorized into sequential grouping cues (those that operate across time, or segregated) and simultaneous grouping cues (those that operate across frequency, or integrated). In addition, schemas (learned patterns) play an important role. Mathematical formalizations to the field of computational auditory perception have also been proposed, for instance, by [Smaragdis \(2001\)](#) and [Cont \(2008\)](#).

The main advantage for humans when transcribing music is our unique capability to identify patterns and our memory, which allows us to predict future events. Using memory in computational transcription usually implies a huge computational cost. It is not a problem to include short term memory, but finding long term repetitions means keeping alive many hypothesis for various frames, which is a very costly task. Solving certain ambiguities that humans can do using long-term memory remains as a challenge. An excellent analysis of prediction, expectation and anticipation of musical events from a psychological point of view is done by [Cont \(2008\)](#), who also proposes computational anticipatory models to address several aspects of musical anticipation. Symbolic music sequences can also be modeled and predicted in some degree ([Paiement et al., 2008](#)), as they are typically composed of repetitive patterns.

Within a short context, a trained musician can not identify any musical information when listening to a 50 ms isolated segment of music. With a 100 ms long segment, some rough pitch estimation can be done, but it is still difficult to identify the instrument. Using longer windows, the timbre becomes apparent. However, multiple f_0 estimation systems that perform the STFT can estimate the pitches using short frames¹. Therefore, probably computers can do a better estimate in isolated time frames, but humans can transcribe better within a wider context.

3.2 Multiple fundamental frequency estimation

Single f_0 estimation methods attempt to find the fundamental frequency in signals with at most one harmonic source sounding at each time. A single harmonic signal can be expressed as the sum of the harmonic part and the residual z :

$$x[n] \approx \sum_{h=1}^H A_h \cos(h\omega_0 n + \phi_h) + z[n] \quad (3.1)$$

where $\omega_0 = 2\pi f_0$. The relation in Eq. 3.1 is approximated for practical use, as the signal can have some harmonic deviations.

A multiple f_0 estimation method assumes that there can be more than one harmonic source in the input signal. Formally, the sum of M harmonic sources can be expressed as:

$$x[n] \approx \sum_{m=1}^M \sum_{h_m=1}^{H_m} A_{m,h_m} \cos(h_m \omega_m n + \phi_m[n]) + \bar{z}[n] \quad (3.2)$$

where $\bar{z}[n]$ is the sum of the residuals of the sources.

It is generally admitted that most single f_0 methods are not adequate for multiple f_0 estimation. A study of the performance of different single f_0 estimation functions applied to the analysis of polyphonic signals can be found in (Yeh, 2008), p. 19.

The problem is that the polyphonic issue is much more complex than the single f_0 estimation. First, the number of sources M must be inferred, in contrast to the single f_0 estimation, where the only decision is to determine whether there is sound or silence. Estimating the number of sounds, or polyphony inference, is a challenging task itself. The noise model is also more complex than in single f_0 signals. In real polyphonic music, besides transients, there can be unpitched sounds (like drums), which usually have a short duration.

¹The typical frame length used to detect multiple f_0 when using the STFT is about 93 ms.

3. MUSIC TRANSCRIPTION

Noise suppression techniques have been proposed in the literature² to allow the subtraction of additive noise from the mixture. And the third major issue is that, besides the source and noise models, in multiple f_0 estimation there is a third model, which is probably the most complex: the interaction between the sources.

For instance, consider two notes playing simultaneously within an octave interval. As their spectrum shows the same harmonic locations than the lowest note playing alone, some other information (such as the energy expected in each harmonic for a particular instrument) is needed to infer the presence of two notes. This issue is usually called octave ambiguity.

According to Klapuri (2004), in contrast to speech, the pitch range is wide³ in music, and the sounds produced by different musical instruments vary a lot in their spectral content. The harmonic pattern of an instrument is also different from low to high notes. And transients and the interference of unpitched content in real music has to be addressed too.

On the other hand, in music the f_0 values are temporally more stable than in speech. Citing Klapuri (2004), it is more difficult to track the f_0 of four simultaneous speakers than to perform music transcription of four-voice vocal music.

As previously discussed in Sec. 2.3.1, consonant intervals are more frequent than dissonant ones in western music. Therefore, pleasant chords include harmonic components of different sounds which coincide in frequency (harmonic overlaps). Harmonic overlaps and beating are the main effects produced by the interaction model, and they are described below.

3.2.1 Harmonic overlap

As pointed out by Klapuri (1998), two sources with fundamental frequencies f_a and f_b are harmonically related when $f_a = \frac{m}{n} f_b$, being m and n positive integer numbers. In this case, every n^{th} harmonic of the source a overlaps every m^{th} of the source b . As mentioned above, this scenario is very frequent in music.

When two sounds are superposed, the corresponding wave functions are summed. When there is a harmonic overlap, two simple harmonic motions with the same frequency, but different amplitudes and phases are added. This produces another simple harmonic motion with the same frequency but different amplitude and phase. Therefore, when two harmonics are overlapped, two sinusoids of the same frequency are summed in the waveform, resulting a signal with the same frequency and which magnitude depends on their phase difference.

Considering K sinusoids overlap, the resulting sinusoid can be written as:

²For a review of noise estimation and suppression methods, see (Yeh, 2008), chapter 4.

³The tessitura of an instrument is the f_0 range that it can achieve.

$$A \cos(\omega n + \phi) = \sum_{k=1}^K A_k \cos(\omega n + \phi_k) \quad (3.3)$$

Using trigonometric identity, the resulting amplitude (Yeh and Roebel, 2009) can be calculated as:

$$A = \sqrt{\left[\sum_{k=1}^K A_k \cos(\phi_k) \right]^2 + \left[\sum_{k=1}^K A_k \sin(\phi_k) \right]^2} \quad (3.4)$$

From which the estimated amplitude A of two overlapping partials with the same frequency, different amplitude, and phase difference ϕ_Δ is:

$$A = \sqrt{A_1^2 + A_2^2 + 2A_1A_2 \cos(\phi_\Delta)} \quad (3.5)$$

As pointed out by Yeh and Roebel (2009), two assumptions are usually made for analyzing overlapping partials: the additivity of linear spectrum and the additivity of power spectrum. The additivity of linear spectrum $A = A_1 + A_2$ assume that the two sinusoids are in phase, i.e., $\cos(\phi_\Delta) = 1$. The additivity of power spectrum $A = \sqrt{A_1^2 + A_2^2}$ is true when $\cos(\phi_\Delta) = 0$.

According to Klapuri (2003a), if one of the partial amplitudes is significantly greater than the other, as is usually the case, A approaches the maximum of the two. Looking at Eq. 3.5, this assumption is closely related to the additivity of power spectrum, which experimentally (see Yeh and Roebel, 2009) obtains better amplitude estimates than considering $\cos(\phi_\Delta) = 1$.

Recently, Yeh and Roebel (2009) proposed an expected overlap model to get a better estimation of the amplitude when two partials overlap, assuming that the phase difference is uniformly distributed.

3.2.2 Beating

If two harmonics of different sources do not coincide in frequency⁴, but they have similar amplitude and small frequency difference, interference beats can be perceived (see Fig. 3.1).

As pointed out by Wood (2008), p. 158, the physical explanation of dissonance is that we hear unpleasant beats. Beats are periodic variations of loudness, and the frequency of the beats depend on the frequency difference of the two tones.

Even when the frequency difference between two partials is not small enough to produce a perceptible beating, some amount of beating is always produced between a pair of harmonics, even when they belong to the same source. The

⁴For instance, due to slight harmonic deviations.

3. MUSIC TRANSCRIPTION

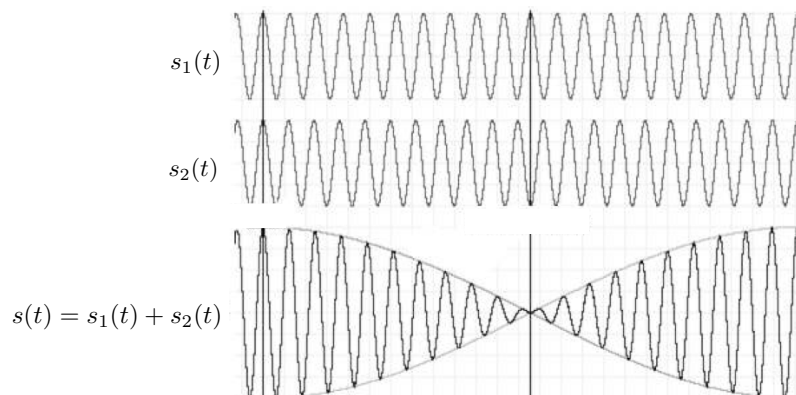


Figure 3.1: Interference tones of two sinusoidal signals of close frequencies. Fig. extracted from <http://www.phys.unsw.edu.au/jw/beats.html>

beating effect generates spectral components not belonging to any original source (see Fig. 3.2), producing ghost fundamental frequencies, and it also alters the original partial amplitudes in the spectrum.

3.2.3 Evaluation metrics

There are a number of evaluation issues for multiple f_0 estimation. First, it is difficult to get real data aligned with the ground-truth pitches, and doing this alignment by hand is a time-consuming task. A simple database can be built synthesizing data from MIDI files, but synthesized data is not equivalent to real sounds.

One of the reason for this difference is reverberation. Reverberation, or reverb, is created when a sound is produced in a closed space, causing a large number of echoes to build up and then slowly decaying as the sound is absorbed by the walls and air (Lloyd, 1970). Reverberation overlaps preceding sounds with following ones. As pointed out by Beauchamp et al. (1993) and Yeh et al. (2006), when a monophonic recording is carried out in a reverberant environment, the recorded signal can become polyphonic. Citing Yeh (2008), the reverberated parts are quite non-stationary, increasing the complexity of the analysis. Besides reverb, other musical production effects like chorus or echo, which are common in commercial recordings, can complicate the analysis.

Other difference is that synthesized signals from MIDI files typically have constant envelopes, making unnatural sounds for instruments with varying envelopes like, for instance, a sax or a violin. Real sounds are usually much more variable and less static than synthesized sounds.

3.2. MULTIPLE FUNDAMENTAL FREQUENCY ESTIMATION

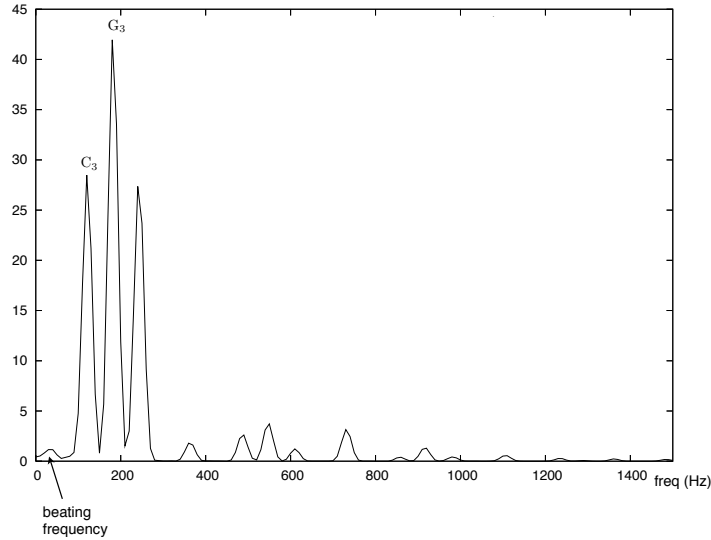


Figure 3.2: Example spectrum of two piano sounds with fundamental frequencies C_3 (130.81 Hz) and G_3 (196 Hz). A beating component appears at frequency 65 Hz, corresponding to a C_2 ghost pitch.

Different metrics have been used in the literature to evaluate polyphonic estimation methods. They can be classified into frame-by-frame metrics (fundamental frequencies are evaluated within single frames), and note-based metrics (note onsets and durations are also taken into account). The former are used to evaluate most multiple f_0 estimation methods, whereas note-based metrics are suitable for the evaluation of those approaches that also perform f_0 tracking⁵.

Frame-based evaluation

Within a frame level, a false positive (FP) is a detected pitch which is not present in the signal, and a false negative (FN) is a missing pitch. Correctly detected pitches (OK) are those estimated pitches that are also present in the ground-truth.

A commonly used metric for frame-based evaluation is the accuracy, which can be defined as:

$$Acc = \frac{\Sigma_{OK}}{\Sigma_{OK} + \Sigma_{FP} + \Sigma_{FN}} \quad (3.6)$$

⁵This term refers to the tracking of the f_0 estimates along consecutive frames in order to add a temporal continuity to the detection.

3. MUSIC TRANSCRIPTION

Alternatively, the error metric can be defined in precision/recall terms. The precision is the percentage of the detected pitches that are correct, whereas the recall is the percentage of the true pitches that were found with respect to the actual pitches.

$$\text{Prec} = \frac{\Sigma_{OK}}{\Sigma_{OK} + \Sigma_{FP}} \quad (3.7)$$

$$\text{Rec} = \frac{\Sigma_{OK}}{\Sigma_{OK} + \Sigma_{FN}} \quad (3.8)$$

The balance between precision and recall, or F-measure, which is commonly used in string comparison, can be computed as:

$$\text{F-measure} = \frac{2 \cdot \text{Prec} \cdot \text{Rec}}{\text{Prec} + \text{Rec}} = \frac{\Sigma_{OK}}{\Sigma_{OK} + \frac{1}{2}\Sigma_{FP} + \frac{1}{2}\Sigma_{FN}} \quad (3.9)$$

Precision, recall and F-measure can also be expressed in percentages, multiplying by 100 the expressions above. Note that F-measure yields higher values than accuracy having the same amount of errors.

An alternative metric based on the “speaker diarization error score” from the NIST⁶ was proposed by Poliner and Ellis (2007a) to evaluate frame-level polyphonic transcriptions. The NIST metric consists of a single error score which takes into account substitution errors (mislabeling an active voice, E_{subs}), miss errors (when a voice is truly active but results in no transcript, E_{miss}), and false alarm errors (when an active voice is reported without any underlying source, E_{fa}).

This metric was proposed to avoid counting errors twice as accuracy of F-measure do in some situations. For instance, using accuracy, if there is a C₃ pitch in the reference ground-truth but the system reports a C₄, two errors (a false positive and a false negative) are counted. However, if no pitch was detected, only one error would be reported.

To compute the total error (E_{tot}) in T frames, the estimated pitches at every frame are denoted as N_{sys} , the ground-truth pitches as N_{ref} , and the number of correctly detected pitches as N_{corr} , which is the intersection between N_{sys} and N_{ref} .

$$E_{tot} = \frac{\sum_{t=1}^T \max(N_{ref}(t), N_{sys}(t)) - N_{corr}(t)}{\sum_{t=1}^T N_{ref}(t)} \quad (3.10)$$

The substitution, miss and false alarm errors are defined as follows:

$$E_{subs} = \frac{\sum_{t=1}^T \min(N_{ref}(t), N_{sys}(t)) - N_{corr}(t)}{\sum_{t=1}^T N_{ref}(t)} \quad (3.11)$$

⁶National Institute of Standards and Technology.

$$E_{miss} = \frac{\sum_{t=1}^T \max(0, N_{ref}(t) - N_{sys}(t))}{\sum_{t=1}^T N_{ref}(t)} \quad (3.12)$$

$$E_{fa} = \frac{\sum_{t=1}^T \max(0, N_{sys}(t) - N_{ref}(t))}{\sum_{t=1}^T N_{ref}(t)} \quad (3.13)$$

Poliner and Ellis (2007a) suggest that, as in the universal practice in the speech recognition community, this is probably the most adequate measure, since it gives a direct feel for the quantity of errors that will occur as a proportion of the total quantity of notes present.

To summarize, three alternative metrics are used in the literature to evaluate multiple f_0 estimation systems within a frame level: accuracy (Eq. 3.6), F-measure (Eq. 3.9), and total error (Eq. 3.10).

The accuracy is the most widely used metric for frame by frame evaluation. The main reason for using accuracy instead of F-measure is that an equilibrate balance between precision and recall is probably less adequate for this task. Typically, multiple f_0 estimation methods obtain higher precision than recall. This occurs because some analyzed mixtures contain many pitches with overlapped harmonics that can be masked by the other components. An experiment was carried out by Huron (1989) to study the limitations in listeners abilities to identify the number of concurrent sounding voices. The most frequent type of confusion was the underestimation of the number of sounds. Some pitches can be present in the signal, but they can become almost unhearable, and they are also very difficult to detect analytically. For instance, when trying to listen an isolated 93 ms frame with 6 simultaneous pitches, we usually tend to underestimate the number of sources.

Note-based evaluation

Instead of counting the errors at each frame and summing the result for all the frames, alternative metrics have been proposed to evaluate the temporal continuity of the estimate. Precision, recall, F-measure and accuracy are also frequently used for note level evaluation. However, it is not trivial to define what is a correctly detected note, a false positive, and a false negative.

The note-based metric proposed by Ryyänen and Klapuri (2005) considers that a reference note is correctly transcribed when their pitches are equal, the absolute difference between their onset times is smaller than a given onset interval, and the transcribed note is not already associated with another reference note. Results are reported using precision, recall, and the mean overlap ratio, which measures the degree of temporal overlap between the reference and transcribed notes.

3. MUSIC TRANSCRIPTION

The multiple f_0 tracking evaluation metric used in [MIREX \(2007, 2008\)](#) reports the results in two different ways. In the first setup, a returned note is assumed correct if its onset is within ± 50 ms of a reference note and its f_0 is correct, ignoring the offset values. In the second setup, on top of the above requirements, a correct note was required to have an offset value within 20% of the reference note duration around the reference offset, or within 50 ms.

[Daniel et al. \(2008\)](#) introduced a perceptual error metric for multiple f_0 evaluation. To estimate the perceptual relevance of note insertions, deletions, replacement, note doubling, etc., thirty-seven subjects were asked to obtain a subjective scale of discomfort for typical errors. Then, a set of weighting coefficients related to the errors made when detecting a wrong octave, fifth, other intervals, deletion, duration, and onset, were estimated to get a perceptive F-measure. In this metric, octave errors have a lower impact than fourth errors, for instance.

The perceptual measure proposed by [Fonseca and Ferreira \(2009\)](#) for note-based evaluation considers different metrics for decay (percussive) and sustained sounds. The decay-based detection only takes into account onset and pitch, whereas the sustained sounds are evaluated by overlapping the original and transcribed piano-rolls with a tolerance degree (i.e., considering pitch, onset and offset). As the nature of the sounds is unknown a-priori, the final score is set as the average of the decay and sustain scores.

3.3 Onset detection

Onset detection refers to the detection of the beginnings of discrete events in audio signals. These events can be either pitched or unpitched sounds. It is a component of the segmentation process which aims to divide a musical signal into smaller units (individual notes, chords, drum sounds, etc.).

This task can be useful for a number of applications. Some tempo estimation and beat tracking methods use the onsets information. Onset times are also useful for multiple f_0 estimation and source separation tasks, as the beginning of notes can be used for segmentation. An onset list can also serve as a robust fingerprint for a song, to identify it in a database. Some music classification approaches, like [Lidy et al., 2007](#), use onsets as input features. Other applications include music editing, as they can be used to divide the song into logical parts, and audio/video synchronization.

Some works define onset as the time instant when the attack interval or when a musical sound begins, whereas others, like [Bello et al., 2005](#), consider onsets as the time instants when a transient starts.

Onsets can be categorized according to the source which produce them. For instance, [Tan et al. \(2009\)](#) classifies onsets as those produced by unpitched

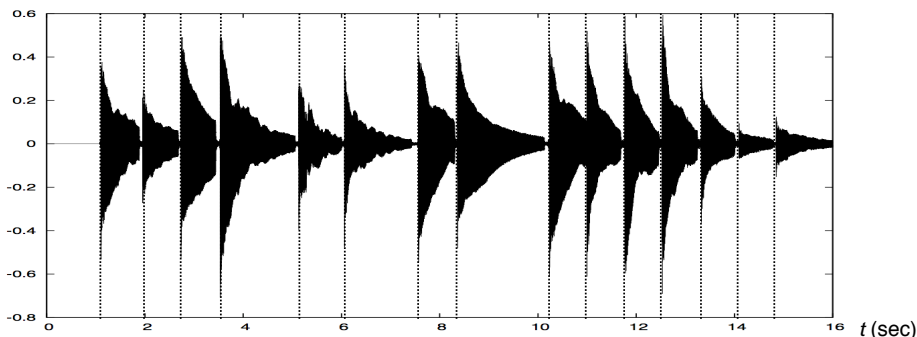


Figure 3.3: Example of a guitar sound waveform. The actual onsets are marked with dashed vertical lines.

sounds like drums, pitched percussive onsets like pianos, and pitched non-percussive onsets like bowed strings. Unpitched and pitched percussive sounds produce hard onsets, whereas pitched non-percussive timbres usually generate soft onsets.

3.3.1 Evaluation metrics

Rigorous evaluation of onset detection is a complex task (Rodet et al., 2004). The evaluation results of onset detection algorithms are in some cases not comparable, as they depend much on the database used for the experiments.

According to Moore (1997), the human ear can not distinguish between two transients less than 10 ms apart. However, as Bello et al. (2005) points, correct matches in the evaluation usually imply that the target and detected onsets are within a 50 ms window, to consider the inaccuracy of the hand labelling process.

In the literature, there is agreement to express the onset detection results in precision/recall and F-measure terms, similarly to Eqs. 3.7, 3.8, and 3.9 used for multiple f_0 estimation. A false positive (FP) is a detected onset that was not present in the signal, and a false negative (FN) is an undetected onset. The precision is the ratio of the correctly detected onsets, and the recall is the ratio between the true onsets that were found with respect to the ground-truth onsets.

4

State of the art

This chapter presents an overview of previous studies for single f_0 estimation, followed by a deeper review and discussion of different multiple f_0 estimation and onset detection methods.

4.1 Single fundamental frequency estimation

A number of single fundamental frequency estimation systems have been developed in the time and the frequency domains. For a review of different methods, see (Rabiner et al., 1976), (Hess, 1983), and (Gerhard, 2003). Most of the proposed approaches were initially developed for speech signals¹, but they have also been successfully applied to musical harmonic sounds.

4.1.1 Time domain methods

Time domain methods look for a repetitive pattern in the signal, corresponding to the fundamental period. A widely used technique is the autocorrelation function, which is defined for a signal $x[t]$ with a frame length W as:

$$\text{ACF}_x[\tau] = \sum_{k=t}^{t+W-1} x[k]x[k + \tau] \quad (4.1)$$

where τ represents the lag value. The peaks of this function correspond to multiples of the fundamental period. Usually, autocorrelation methods select the highest non-zero lag peak over a given threshold within a range of lags. However, this technique is sensitive to formant structures, producing octave errors. As Hess (1983) points out, some methods like center clipping (Dubnowski

¹Detecting the fundamental frequency in speech signals is useful, for instance, for prosody analysis. Prosody refers to the rhythm, stress, and intonation of connected speech.

4. STATE OF THE ART

et al., 1976), or spectral flattening (Sondhi, 1968) can be used to attenuate these effects.

The squared difference function (SDF) is a similar approach to measure dissimilarities, and it has been used by de Cheveigné and Kawahara (2002) for the YIN algorithm.

$$\text{SDF}_x[\tau] = \sum_{k=t}^{t+W-1} (x[k] - x[k + \tau])^2 \quad (4.2)$$

The YIN method computes the cumulative mean normalized difference function (SDF'), which is obtained by dividing each value of the SDF by its average over shorter-lag values.

$$\text{SDF}'_x[\tau] = \begin{cases} 1 & \text{if } \tau = 0, \\ \frac{\text{SDF}_x[\tau]}{(1/\tau) \sum_{j=1}^{\tau} \text{SDF}_x[j]} & \text{otherwise} \end{cases} \quad (4.3)$$

The main advantage of using the SDF' function is that it tends to remain large at low lags, dropping below 1 only where SDF falls below the average. Basically, it removes dips and lags near zero avoiding super-harmonic errors, and normalization makes the function independent of the absolute signal level.

An absolute threshold is set, choosing the first local minimum of SDF' below that threshold. If none is found, the global minimum is chosen instead. Once the lag value τ is selected, a parabolic interpolation of immediate neighbors is done to increase the accuracy of the estimate, obtaining τ' , and the detected fundamental frequency is finally set as $f_0 = f_s/\tau'$.

YIN is a robust and reliable algorithm that have been successfully used as basis for singing voice transcription methods, like the one proposed by Ryyänänen and Klapuri (2004).

4.1.2 Frequency domain methods

Usually, methods in the frequency domain analyze the locations or the distance between hypothetical partials in the spectrum.

Cepstrum

The real cepstrum of a signal is the inverse Fourier transform of the logarithm of the magnitude spectrum.

$$\text{CEP}_x[\tau] = \text{IDFT}\{\log(|\text{DFT}(x[n])|)\} \quad (4.4)$$

It was introduced for fundamental frequency estimation of speech signals by Noll and Schroeder (1964), who gave a complete methodology in (Noll, 1967).

Since temporal periodicities in a speech signal cause periodic ripples in the amplitude spectrum, the cepstrum gives the frequency of the ripple, which is inversely proportional to the fundamental frequency of the speech.

ACF-based fundamental frequency estimators have some similarities with cepstrum-based methods. The ACF of a time-domain signal $x[n]$ can also be expressed using the DFT :

$$\text{ACF}_x[\tau] = \text{IDFT}\{|\text{DFT}(x[n])|^2\} = \frac{1}{K} \sum_{k=0}^{K-1} \left[\cos\left(\frac{2\pi\tau k}{K}\right) |X[k]|^2 \right] \quad (4.5)$$

Note that the cosine factor emphasizes the partial amplitudes at those harmonic positions multiple of τ . The main difference between autocorrelation and cepstrum is that autocorrelation uses the square of the DFT, and the cepstrum performs the logarithm. Squaring the DFT causes to raise spectral peaks but also the noise. Using the logarithm flats the spectrum, reducing noise but also the harmonic amplitudes.

Therefore, as pointed out by [Rabiner et al. \(1976\)](#), the cepstrum performs a dynamic compression over the spectrum, flattening unwanted components and increasing the robustness for formants, but rising the noise level, whereas autocorrelation emphasizes spectral peaks in relation to noise, but raising the strength of spurious components.

Both ACF and cepstrum-based methods can be classified as spectral location f_0 estimators.

Spectral autocorrelation

The main drawback of the spectral location f_0 estimators is that they are very sensitive to harmonic deviations from their ideal position. Some methods, like the one proposed by [Lahat et al. \(1987\)](#), perform autocorrelation over the spectrum.

$$\text{ACFS}_X[\tau] = \frac{2}{K} \sum_{k=0}^{K/2-\tau-1} |X[k]| |X[k+\tau]| \quad (4.6)$$

The maximum ACFS value is usually found when $\tau = f_0(K/f_s)$. Spectral autocorrelation is more robust against inharmonicity than ACF, as the spectrum can be shifted without affecting the detection results.

Methods using spectral autocorrelation can be classified as a spectral interval f_0 estimators, as they look for harmonic intervals rather than their locations.

4. STATE OF THE ART

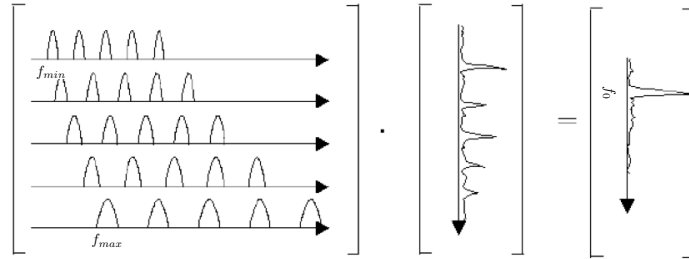


Figure 4.1: Maximum likelihood from [Noll \(1969\)](#).

Pattern matching in the frequency domain

Other approaches compare the measured spectrum with a given harmonic spectral pattern. The spectral pattern can be either a specific spectral model or a sequence of equally spaced components, which is often called a harmonic comb. A comb filter is composed by a set of equally spaced pass-bands. In the case of the optimum comb filter algorithm, the location of the passbands coincide with the harmonic locations.

[Brown \(1992\)](#) uses a constant-Q transform in a preprocessing stage to correlate the spectrum with an ideal harmonic pattern, which consists of ones at the positions of harmonic frequency components. The f_0 is estimated by looking at the position of the best approximation to the ideal pattern, which corresponds to the greatest cross-correlation peak.

The Maximum likelihood² (ML) algorithm proposed by [Noll \(1969\)](#) searches through a set of possible ideal spectra and chooses the one which best matches the shape of the input spectrum (see [Fig. 4.1](#)). The system is suitable for tuned pitched sounds, as the base frequencies of ideal spectra coincide with the musical pitches.

The method proposed by [Doval and Rodet \(1993\)](#) can be also placed in the harmonic matching class, as it constructs a histogram on the selected intervals by computing the value of the likelihood function for each interval.

An interesting pattern matching approach is the two-way mismatch (TWM) algorithm from [Maher \(1990\)](#); [Maher and Beauchamp \(1994\)](#). It is based on the comparison between the partials obtained from the STFT and predicted sequences of harmonics relative to the f_0 . The differences between the measured and predicted partials are referred as the mismatch error, which is calculated in two ways, as shown in [Fig. 4.2](#). The first measures the frequency difference

²Citing ([Davy, 2006a](#)), the likelihood can be seen as a similarity measure between the signal and the model, via its parameters. The expectation maximization (EM) algorithm by [Moon \(1996\)](#) can be used for maximum likelihood parameter estimation.

4.1. SINGLE FUNDAMENTAL FREQUENCY ESTIMATION

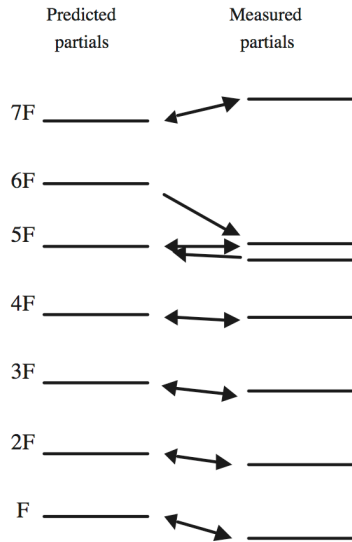


Figure 4.2: Two way mismatch procedure from [Maher and Beauchamp \(1994\)](#).

between each measured partial and its nearest harmonic neighbor in the predicted sequence, whereas the second measures the mismatch between each predicted harmonic and its nearest neighbor in the measured sequence. Each match is weighted by the amplitudes of the observed peaks. This method tries to reduce octave errors, applying a penalty to missing and extra harmonics relative to the predicted pattern. The methodology was also used for duet³ separation⁴.

[Cano \(1998\)](#) introduced some modifications over the TWM to improve the original SMS analysis developed by [Serra \(1997\)](#). These modifications include a pitch dependent analysis window using adaptive window length, a more restrictive selection of spectral peaks to be considered, f_0 tracking using short-term history to choose between candidates with similar TWM error, to restrict the frequency range of possible candidates, and to discriminate between pitched and unpitched parts.

4.1.3 Perceptual models

Some f_0 estimation methods use perceptual models of human hearing. The unitary model of the auditory system, introduced by [Meddis and Hewitt \(1991a,b\)](#), can estimate the f_0 of a signal by measuring the periodicity of

³A duet is composed by two melodies playing simultaneously.

⁴For this reason, this algorithm could also be considered as a multiple f_0 estimation system belonging to the joint estimation methods category (see Sec. 4.2.3).

4. STATE OF THE ART

the time-domain amplitude envelope. This model represent a tradeoff between spectral location and spectral interval methods.

As pointed out by Klapuri (2003a), time domain periodicity analysis methods are prone to errors in f_0 halving, whereas frequency domain methods are prone to errors in f_0 doubling. This is because the time-domain signal is periodic at half the f_0 (twice the fundamental period), whereas the spectrum is periodic at double the f_0 rate. The unitary model gets a good compromise between both.

As pointed out by Klapuri and Astola (2002), the unitary model is widely accepted as a psychoacoustically valid mid-level representation. To compute it (see (Meddis and Hewitt, 1991a) and (Duda et al., 1990) for details), a cochlear frequency analysis is first done using an auditory filter-bank. Then, a simulation of the hair cells⁵ analysis is performed through half-wave rectification⁶, compression and low-pass filtering of the signals at each frequency channel. The periodicity at each channel is estimated through the ACF function, yielding a correlogram, which is a three-dimensional representation of time, frequency and ACF lag. Finally, the summary autocorrelation function (SACF), is computed, summing the ACF across channels. The greatest value of the SACF function is used as an indicator of the perceived pitch.

Computationally efficient implementations of the SACF have been proposed by Tolonen and Karjalainen (2000), and Klapuri and Astola (2002). For a better coverage about pitch perception models, see the review from de Cheveigné (2005).

4.1.4 Probabilistic models

As pointed out by Roads (1996), a small time window is often not enough for a human to identify pitch, but when many frames are played one after another, a sensation of pitch becomes apparent. This is the main motivation to introduce probabilistic models for f_0 tracking, which can be used to refine the f_0 estimate.

Intuitively, a simple f_0 tracking approach would consist in giving preference to f_0 hypotheses that are close to the hypothesis of the last time frame. A more reliable method is to use statistical models, like hidden Markov models (HMMs), which track variables through time. HMMs are state machines, with a hypothesis available for the output variable at each state. At each time frame, the HMM moves from the current state to the most likely next state, based on the input to the model and the state history which is represented in the current state.

⁵Hair cells convert cochlear movements into neural impulses.

⁶The half-wave rectification of a time-domain signal keeps the positive values, zeroing the negative ones.

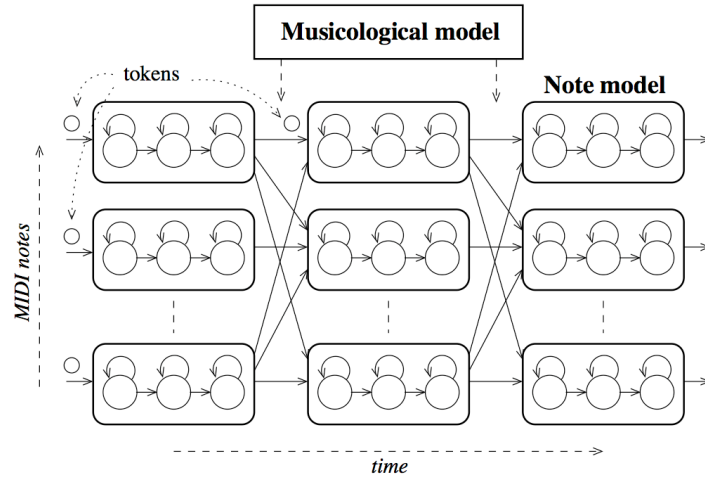


Figure 4.3: Combinations of note models and the musicological model from Ryyänen and Klapuri (2004).

A HMM consists of a number of states, the state transition probabilities, the observation likelihood distributions and the initial and final state probabilities. The state transition probabilities of a HMMs are learned in an unsupervised way using the Baum-Welch algorithm (see (Rabiner, 1989) for details). Once the parameters have been estimated, the state sequence that maximizes the probability of the observed data can be efficiently computed using the Viterbi algorithm, described by Viterbi (1967).

HMMs have been used by Ryyänen and Klapuri (2004) for a singing voice transcription system. First, the fundamental frequency, voicing, accent and metrical accent are extracted from the audio signal. The fundamental frequency and voicing are estimated using the YIN algorithm (voicing corresponds to SDF'_x , see Eq. 4.3), and the f_0 is rounded to that of its nearest MIDI pitch. Then, two probabilistic models are used: a note event model and a musicological model. Note events are described using a HMM for each pitch. The HMMs are composed of three states (which can be intuitively identified with the attack, sustain and silence stages), and their inputs are the extracted features (pitch difference⁷, voicing, accent and metrical accent). The musicological model weights transitions between notes using note n-gram probabilities which are dependent on the key of the song, which is estimated using the algorithm proposed by Viitaniemi et al. (2003).

⁷The frequency difference in semitones between the measured f_0 and the nominal pitch of the modeled note.

Finally, the two models are combined into a network (see Fig. 4.3), and the most probable path is found according to the likelihoods given by the note models and the musicological model. The system obtained half amount of errors than the simple f_0 estimation rounded to MIDI pitch, proving the capability of probabilistic models for this task.

4.2 Multiple fundamental frequency estimation

Since the first polyphonic transcription attempt from Moorer (1975) to transcribe duets with some interval constraints, many different techniques have been proposed for multiple f_0 estimation. As pointed out by Klapuri (2004), it is difficult to classify these methods using a single taxonomy, as they are very complex and usually combine several processing principles.

For instance, multiple f_0 estimation systems can be categorized according to their mid-level representation (time-domain, STFT, wavelets, auditory filter banks, etc.), but also to their scope (some methods need a-priori information about the instrument to be transcribed, whereas others can be used to analyze generic harmonic sounds), to their capability for modeling varying timbres (for instance, statistical parametric methods can model varying time-frequency envelopes like those of sax sounds, whereas non-parametric methods can only handle fixed spectral patterns, like piano sounds), or by the way they can estimate the interactions between sources (iterative and joint estimation methods).

In this work, the proposed categorization is based on the core methodology, rather than the mid-level representation used in the single f_0 estimation taxonomy. Existing approaches have been classified into salience functions, iterative cancellation, joint estimation, supervised learning, unsupervised learning, matching pursuit, Bayesian models, statistical spectral models, blackboard systems, and database matching methods. An analysis of the strengths and limitations for each of these categories is done in Sec. 4.3.

4.2.1 Salience methods

Salience methods try to emphasize the underlying fundamental frequencies by applying signal processing transformations to the input signal. In these approaches the core of the detection process relies on the mid-level representation.

Tolonen and Karjalainen (2000) perform the SACF over the signal using an auditory filter bank. The SACF is processed to remove the near-zero lag components and the peaks that are multiple of the other peaks found. The resulting function is called enhanced summary autocorrelation function (ESACF).

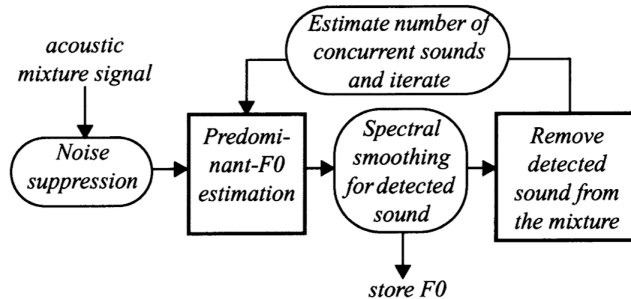


Figure 4.4: Iterative cancellation method from Klapuri (2003a).

The method from Peeters (2006) combines a temporal representation (time-domain ACF and real cepstrum) with a spectral representation (spectral autocorrelation) to reduce octave ambiguities. The best results were reported when combining the spectral autocorrelation function with the cepstrum.

Zhou et al. (2009) propose an efficient method which relies on a novel time-frequency representation called resonator time-frequency image (RTFI). The RTFI (Zhou and Mattavelli, 2007) selects a first order complex resonator filter bank to implement a frequency dependent time-frequency analysis. Harmonic components are extracted by transforming the RTFI average energy spectrum into a relative energy spectrum. Then, a preliminary estimation of pitch candidates is done by converting the RTFI average spectrum into a pitch energy spectrum (PES) and a relative pitch energy spectrum (RPES). The information about harmonic components and pitch candidates are combined to remove extra pitches. Finally, the remaining candidates are filtered out by using a smoothness criterion to remove extra pitches again, considering only cases for which the frequency ratio of two candidates is 2, 3 or 4.

4.2.2 Iterative cancellation methods

Some methods estimate the most prominent f_0 , subtracting it from the mixture and repeating the process for the residual signal until a termination criterion.

In the method proposed by Klapuri (2003a), Fig. 4.4, the spectrum of the signal is warped on a logarithmic frequency scale to compress the spectral magnitudes and remove the noise. The processed spectrum is analyzed into a $2/3$ octave filter bank, and f_0 weights are computed for each band according to the normalized sum of their partial amplitudes. The results are combined by summing the squared band-wise weights, taking inharmonicity (Eq. 2.17) into account. The spectral components of the fundamental frequencies that have the highest global weights are smoothed using the algorithm described in (Klapuri,

4. STATE OF THE ART

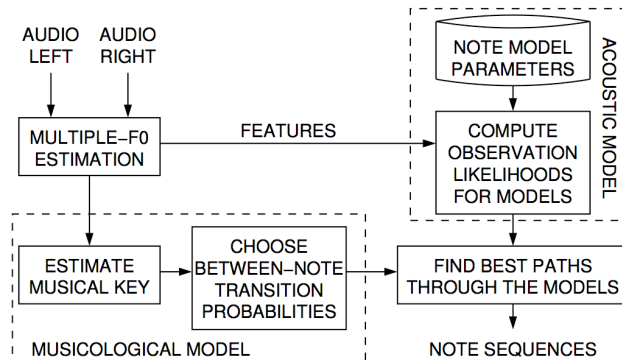


Figure 4.5: Probabilistic framework from Ryyänen and Klapuri (2005).

2001) before subtracting them from the mixture. The weights of each candidate are calculated again after smoothing, and the highest recalculated global weight determines the resulting f_0 . The process stops when the maximum weight related to the signal-to-noise ratio (SNR) is below a fixed threshold.

Klapuri (2005) proposed an alternative method using an auditory filter bank. The signal at each subband is compressed, half-wave rectified and low-pass filtered. Then, similarly to the SACF, the results are combined across channels, but in this method magnitude spectra are summed across channels to obtain a summary spectrum. The most salient f_0 is computed using an approximated $1/h$ spectral envelope model⁸, to remove the source for the mixture while keeping in the residual most of the energy of higher partials. This model is similar to (Klapuri, 2006b), where the input signal is flattened (whitened) to reduce timbral-dependant information, and the salience for each f_0 candidate is computed as a $1/h$ weighted sum of its partials. This same partial weighting scheme is performed by Klapuri (2008) using a computationally efficient auditory model.

The system introduced by Ryyänen and Klapuri (2005) embeds the multiple f_0 estimator from Klapuri (2005) into a probabilistic framework (see Fig. 4.5), similarly to the method from Ryyänen and Klapuri (2004) for singing transcription. As in the latter work, a note event model and a musicological model, plus a silence model are used, and note events are described using a HMM for each pitch. The HMM inputs are the pitch difference between the measured f_0 and the nominal pitch of the modeled note, the f_0 salience, and the onset strength (positive changes in the estimated strengths of f_0 values). The musicological model controls transitions between note HMMs and the silence model using note bigram probabilities which are dependent on the estimated

⁸Spectral amplitudes decrease according to the partial number h , like in a sawtooth signal.

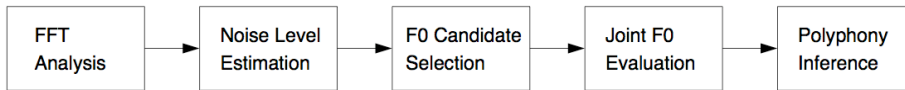


Figure 4.6: Overview of the joint estimation method from Yeh (2008).

key. Like in (Ryynänen and Klapuri, 2004), the acoustic and musicological models are combined into a network which optimal path is found using the token-passing algorithm from Young et al. (1989).

Other examples of iterative cancellation methods are those proposed by Wan et al. (2005), Yin et al. (2005), and Cao et al. (2007).

4.2.3 Joint estimation methods

These methods evaluate a set of possible hypotheses, consisting of f_0 combinations, to select the best one without corrupting the residual at each iteration.

Time-domain methods for joint cancellation of multiple f_0 hypothesis have been proposed by de Cheveigné (1993, 2005). The hypotheses are cancelled using a cascade of filters, and the combination selected is the one that minimizes the residual. In the experiments done by de Cheveigné (2005), different iterative cancellation methods are compared with the joint approach, showing that joint cancellation outperforms the iterative cancellation results.

The method proposed by Yeh (2008) evaluates a set of multiple f_0 hypotheses without cancellation. An adaptive noise level estimation (see Fig. 4.6) is first done using the algorithm described in (Yeh and Roebel, 2006), in order to extract only the sinusoidal components. Then, given a known number of sources, the fundamental frequencies are obtained using the method described in (Yeh et al., 2005). At each spectral frame, to reduce the computational cost, a set of f_0 candidates are selected from the spectral peaks using a harmonic matching technique⁹. Each f_0 hypothesis is related to a hypothetical partial sequence (HPS). The HPS is a source model with estimated frequencies and amplitudes obtained by partial selection and overlapping partial treatment. Partials are identified with spectral peaks within a tolerance deviation from their ideal position. In case that more than one peak is in the tolerance range, the peak forming a smoother HPS envelope is selected. The amplitudes of overlapped partials in the combination are estimated by using linear interpolation, similarly to (Maher, 1990), and a set of rules. Finally, HPS are flattened by exponential compression.

Once HPS are estimated, a score function for a given hypothesis is calculated taking into account, for each hypothetical source, the harmonicity, the

⁹Different methods for selecting the candidates are analyzed in (Yeh, 2008).

4. STATE OF THE ART

smoothness of the spectral envelope, and the synchronous amplitude evolution of the partials. The harmonicity is estimated by measuring the degree of partial deviations weighted by their amplitudes. The spectral smoothness is computed using the mean bandwidth of the FT of the HPS. The spectral centroid is also considered, favoring those HPS with high energy in the lower partials. Finally, the synchronicity is measured as the standard deviation of the mean time¹⁰ of the HPS partials. The score function is a weighted sum of these four criteria, and the weights are optimized using the evolutionary algorithm from Schwefel (1995) with a large dataset.

The estimation of the number of concurrent sounds is finally done by iterative score improvement (Chang et al., 2008; Yeh et al., 2006), based on the explained energy and the improvement of the spectral smoothness. Finally, a postprocessing stage can be added by tracking the f_0 candidates trajectories, using a high-order HMM and a forward-backward tracking scheme proposed by Chang et al. (2008).

The joint estimation approach proposed by Emiya et al. (2008b) detects the onsets in the signal and, for each segment between consecutive onsets, a set of candidates are selected. Then, the most likely combination of pitches within each segment is selected using HMM for tracking candidates and a spectral maximum likelihood method for joint pitch estimation.

The method from Cañadas-Quesada et al. (2009) selects the possible partials from the magnitude spectral peaks using an adaptive logarithmic threshold. The f_0 candidates are then chosen from the selected partials within a given pitch range. For each candidate, a harmonic pattern is built in the log-frequency domain, considering one semitone bandwidth for partial search. Then, all the possible candidate combinations are generated and explained as a sum of Gaussian mixture models (GMM). The GMM weights are obtained using the non-colliding partial magnitudes and the estimated magnitudes¹¹ of colliding partials. Finally, the combination which explains most of the harmonic content and maximizes the temporal similarity with the previous winner combinations is chosen at each frame.

4.2.4 Supervised learning methods

Supervised learning approaches attempt to assign a class to a test sample. Within this context, the classes are the musical pitches, therefore these methods are constrained to tuned harmonic sounds, as they estimate the pitch instead of the fundamental frequency itself.

¹⁰The mean time is an indication of the center of gravity of signal energy (Cohen, 1995). It can be defined in the frequency domain as the weighted sum of group delays.

¹¹The colliding partials are estimated by linear interpolation of non-colliding neighbors.

4.2. MULTIPLE FUNDAMENTAL FREQUENCY ESTIMATION

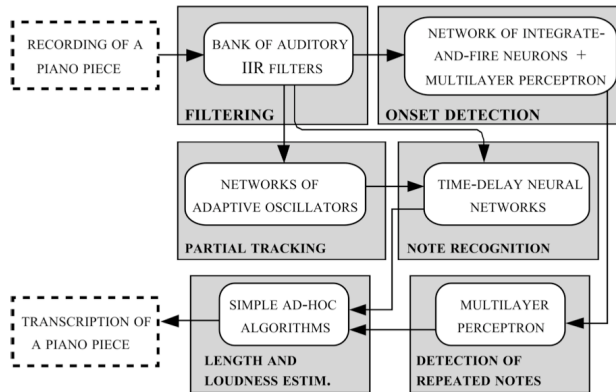


Figure 4.7: SONIC scheme from [Marolt \(2004a\)](#)

The partial tracking method proposed by [Marolt \(2004a,b\)](#) uses a combination of the auditory Patterson-Holdsworth gammatone filterbank with the Meddis hair cell model as a preprocessing stage. Instead of using a correlogram, a modified version of the [Large and Kolen \(1994\)](#) adaptive oscillators is utilized to detect periodicities in output channels of the auditory model. There are 88 oscillators with initial frequencies corresponding to the tuned musical pitches. If the oscillators synchronize with their stimuli (outputs of the auditory model), then the stimuli are periodic, meaning that partials are present in the input signal. This scheme can be used to track partials, even in the presence of vibrato or beating. The model was extended for tracking groups of harmonically related partials, by using the output of the adaptive oscillators as inputs of neural networks. A set of 88 neural networks corresponding to the musical pitches were used, each containing up to 10 oscillators associated to partial frequencies.

The harmonic tracking method from [Marolt \(2004a,b\)](#) was integrated into a system called SONIC (see Fig. 4.7) to transcribe piano music. The combination of the auditory model outputs and the partial tracking neural network outputs are fed into a set of time delay neural networks (TDDN), each one corresponding to a musical pitch. The system also includes an onset detection stage, which is implemented with a fully-connected neural network, and a module to detect repeated notes activations (consecutive notes with the same pitch). The information about the pitch estimate is complemented with the output of the repeated note module, yielding the pitch, length and loudness of each note. The system is constrained to piano transcription, as training samples are piano sounds.

4. STATE OF THE ART

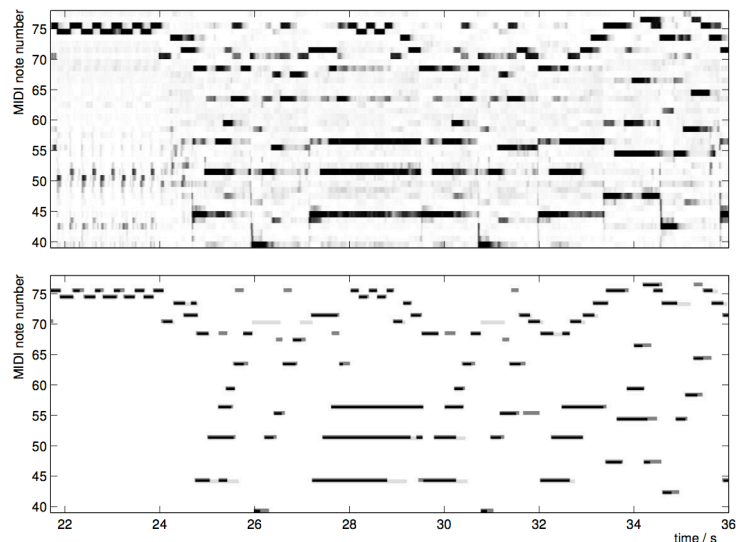


Figure 4.8: HMM smoothed estimation from [Poliner and Ellis \(2007a\)](#) for an excerpt of Für Elise (Beethoven). The posteriorgram (pitch probabilities as a function of time) and the HMM smoothed estimation plotted over the ground-truth labels (light gray) are shown.

[Reis et al. \(2008c\)](#) use genetic algorithms¹² for polyphonic piano transcription. Basically, a genetic algorithm consists of a set of candidate solutions (individuals, or chromosomes) which evolve through inheritance, selection, mutation and crossover until a termination criterion. At each generation, the quality (fitness) of each chromosome is evaluated, and the best individuals are chosen to keep evolving. Finally, the best chromosome is selected as the solution.

In the method proposed by [Reis et al. \(2008c\)](#), each chromosome corresponds to a sequence of note events, where each note has pitch, onset, duration and intensity. The initialization of the population is based on the observed STFT peaks. The fitness function for an individual is obtained from the comparison of the original STFT with the STFT of synthesized versions of the chromosomes given an instrument. The method is constrained to the a priori knowledge of the instrument to be synthesized. The system was extended in [Reis et al. \(2008b\)](#) by combining the genetic algorithm with a memetic algorithm (gene fragment competition), to improve the quality of the solutions during the evolutionary process.

¹²Genetic algorithms are evolutionary methods based on Darwin natural selection proposed by [Holland \(1992\)](#).

Poliner and Ellis (2007a,b) combine Support Vector Machines (SVMs¹³) in a frame-by-frame spectral analysis with HMMs to refine the estimate. The proposed approach trains a SVM with spectrograms of synthesized MIDI files. 87 binary note classifiers detect the presence of the notes at a given frame. Each frame is represented by a 255-element feature vector consisting on normalized spectral bins. The SVM classification output (called ‘posteriorgram’) is parsed through a two state (on/off) HMM to increase the temporal coherence in the pitch estimate. The HMM achieves temporal smoothing (see Fig. 4.8) by estimating the probability of seeing a particular classifier label given a true pitch state with the likelihood of each note being ‘on’ according to the classifiers output. The system is intended for polyphonic piano transcription, experimentally outperforming the Marolt (2004a,b) results.

SVMs have been also used by Zhou (2006) (Method II), using 88 binary classifiers, each corresponding to one pitch. The classifier inputs are the peaks extracted from the RTFI energy spectrum.

4.2.5 Unsupervised learning methods

The goal of non-negative matrix factorization (NMF), first proposed by Lee and Seung (1999), is to approximate a non-negative matrix \mathbf{Y} as a product of two non-negative matrices \mathbf{W} and \mathbf{H} , in such a way that the reconstruction error is minimized:

$$\mathbf{X} \approx \mathbf{W}\mathbf{H} \quad (4.7)$$

This method has been used for music transcription, where typically \mathbf{X} is the spectral data, \mathbf{H} corresponds to the spectral models (basis functions), and \mathbf{W} are the weightings, i.e., the intensity evolution along time (see Fig. 4.9). This methodology is suitable for instruments with a fixed spectral profile¹⁴, such as piano sounds.

There are different ways to design the cost function in order to minimize the residual. For instance, Cont (2006) assumes that the correct solution for a given spectrum uses a minimum of templates, i.e., that the solution has the minimum number of non-zero elements in \mathbf{H} . NMF methods have also been used for music transcription by Plumbley et al. (2002), Smaragdis and Brown (2003),

¹³SVMs are supervised learning methods for classification (see (Burges, 1998)). Viewing input data as sets of vectors in an n-dimensional space, a SVM constructs separating hyperplanes in that space in such a way that the margins between the data sets are maximized.

¹⁴In the scope of this work, an instrument with a fixed spectral profile is referred when two notes of that instrument playing the same pitch produce a very similar sound, as it happens with piano sounds. As an opposite example, a sax can’t be considered to have a fixed spectral profile, as real sax sounds usually contain varying dynamics and expressive alterations, like breathing noise, that do not sound in the same way than other notes with the same pitch.

4.2. MULTIPLE FUNDAMENTAL FREQUENCY ESTIMATION

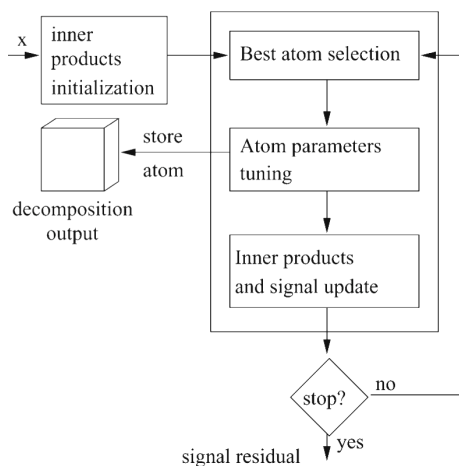


Figure 4.10: Modified MP algorithm from [Leveau et al. \(2008\)](#) for the extraction of harmonic atoms.

that are selected from a dictionary. At the first iteration of the algorithm, the atom which gives the largest inner product with the analyzed signal is chosen. Then, the contribution of this function is subtracted from the signal and the process is repeated on the residue. MP minimizes the residual energy by choosing at each iteration the most correlated atom with the residual. As a result, the signal is represented as a weighted sum of atoms from the dictionary plus a residual.

The method proposed by [Cañadas-Quesada et al. \(2008\)](#) is based on harmonic matching pursuit (HMP) from [Gribonval and Bacry \(2003\)](#). The HMP is an extension of MP with a dictionary composed by harmonic atoms. Within this context, a Gabor atom¹⁶ can be identified with a partial, and a harmonic atom is a linear combination of Gabor atoms (i.e., a spectral pattern). The algorithm from [Cañadas-Quesada et al. \(2008\)](#) extends HMP to avoid inaccurate decomposition when there are overlapped partials, by maximizing the smoothness of the spectral envelope for each harmonic atom. The smoothness maximization algorithm is similar to the one proposed by [Klapuri \(2003a\)](#). The performance of this method when dealing with harmonically related simultaneous notes is further described by [Ruiz-Reyes et al. \(2009\)](#).

[Leveau et al. \(2008\)](#) propose a modified MP algorithm which can be applied to the whole signal, instead of the frame by frame basis. The harmonic atoms extraction method is shown in Fig. 4.10. Molecules are considered as a group of several atoms of the same instrument in successive time windows.

¹⁶Gabor atoms are time-frequency atomic signal decompositions proposed by [Gabor \(1946, 1947\)](#). They are obtained by dilating, translating and modulating a mother generating function.

4.2.7 Bayesian models

Citing [Davy \(2006b\)](#), tonal music can be exploited to build a Bayesian model, that is, a mathematical model embedded into a probabilistic framework that leads to the simplest model that explains a given waveform. Such models are also known as generative models because they can be used to generate data by changing parameters and the noise. Some multiple f_0 estimation systems rely on generative models of the acoustic waveform. Most of these models assume that the fundamental frequency belongs to a fixed grid, associated to the pitches.

The method proposed by [Cemgil et al. \(2003, 2006\)](#) is based on a generative model formulated as a dynamical bayesian network. The probabilistic model assumes harmonic frequency relationships of the partials and exponentially decaying spectrum envelope from one partial to another. This approach allows to write many classical noisy sum-of-sines models into a sequential form. The model relies on sinusoids with damped amplitude and constant frequency. A piano-roll is inferred from the observation, assigning to each of the grid frequencies the state ‘mute’ or ‘sound’ at each instant. The algorithm for estimating the most likely piano-roll is based on EM and Kalman filtering on a sliding window over the audio signal. This can be considered as a time-domain method (DFT is not explicitly calculated), which can be used to analyze music to sample precision, but with a very high computational cost.

[Vincent and Rodet \(2004\)](#) propose a generative model combining a nonlinear Independent Subspace Analysis (ISA¹⁷) and factorial HMM. The method is based on creating specific instrument models based on learning. The spectra of the instrument sounds are modeled by using the means and variances of partial amplitudes, the partial frequencies and the residuals. To transcribe a signal, the spectrum is considered as a sum of spectral models which weights are optimized using the second order Newton method. The HMM is used for adding temporal continuity and modeling note duration priors.

Other Bayesian approaches for music transcription are those proposed by [Kashino and Tanaka \(1993\)](#), [Sterian \(1999\)](#), [Walmsley et al. \(1999\)](#), [Raphael \(2002\)](#), [Kashino and Godsill \(2004\)](#), [Dubois and Davy \(2005, 2007\)](#), [Vincent and Plumbley \(2005\)](#), and [Davy et al. \(2006\)](#). For a review on this topic, see ([Cemgil, 2004](#)) and ([Davy, 2006b](#)).

4.2.8 Statistical spectral models

[Goto \(2000\)](#) describes a method called PreFEst (see Fig. 4.11) to detect melody and bass lines in musical signals. The system assumes that the melody and bass

¹⁷ISA combines the multidimensional ICA with invariant feature extraction. Linear ISA describes the short-time power spectrum of a musical excerpt as a sum of power spectra with time-varying weights, using a Gaussian noise for modeling the error.

4.2. MULTIPLE FUNDAMENTAL FREQUENCY ESTIMATION

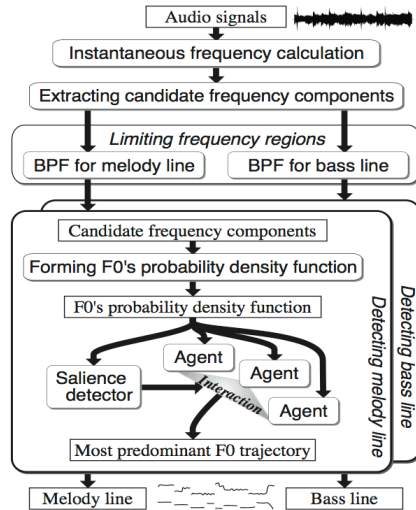


Figure 4.11: Overview of the system proposed by Goto (2000).

are the most predominant harmonic structures in high and low frequency regions respectively. First, the STFT is apportioned through a multirate filterbank, and a set of candidate frequency components are extracted. Then, two bandpass filters are used to separate the spectral components of the bass and melody. For each set of filtered frequency components, the method forms a probability density function (PDF) of the f_0 . The observed PDF is considered as being generated by a weighted mixture of harmonic-structure tone models. The model parameters are estimated using the EM algorithm. To consider a continuity of the f_0 estimate, the most dominant and stable f_0 trajectory is selected, by tracking peak trajectories in the temporal transition of the fundamental frequencies PDFs. To do this, a saliency detector selects salient promising peaks in the PDFs, and agents driven by those peaks track their trajectories. The system works in real time.

Kameoka et al. (2007) propose a method called harmonic temporal structured clustering (HTC). This approach decomposes the power spectrum time series into sequential spectral streams (clusters) corresponding to single sources. This way, the pitch, intensity, onset, duration, and timbre features of each source are jointly estimated. The input of the system is the observed signal, characterized by its power spectrogram with log-frequency. The source model (see Fig. 4.12) assumes smooth power envelopes with decaying partial amplitudes. Using this model, a goodness of the partitioned cluster is calculated using the Kullback-Liebler (KL) divergence. The model parameters are estimated using the expectation-constrained maximization (ECM) algorithm

4. STATE OF THE ART

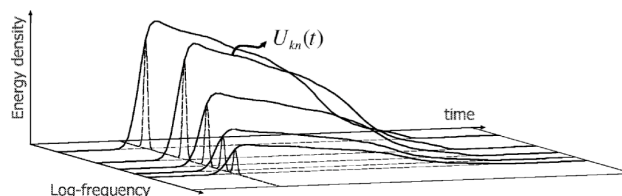


Figure 4.12: HTC spectral model of a single source from [Kameoka et al. \(2007\)](#).

from [Meng and Rubin \(1993\)](#), which is computationally simpler than the EM algorithm. In the evaluation done by [Kameoka et al. \(2007\)](#), the HTC system outperformed the PreFEst results.

The method from [Li and Wang \(2007\)](#) is similar to ([Ryynänen and Klapuri, 2005](#)) in the sense that the preliminary pitch estimate and the musical pitch probability transition are integrated into a HMM. However, for pitch estimation, [Li and Wang \(2007\)](#) use statistical tone models that characterize the spectral shapes of the instruments. Kernel density estimation is used to build the instrument models. The method is intended for single instrument transcription.

4.2.9 Blackboard systems

The blackboard systems, introduced by [Engelmore and Morgan \(1988\)](#), have been also applied to music transcription. The name is based on the idea of a group of experts standing around a blackboard working together to solve a problem. The blackboard is a hierarchical data-space, where the hypotheses are proposed. The experts or knowledge sources typically consist on a set of rules (a sort of if/then conditions). They develop the hypotheses and remove the unsupported ones in the blackboard. The scheduler is the third main component, and it determines the order in which knowledge sources are allowed to act. The system converges when the knowledge sources are satisfied with the hypotheses in the blackboard given an error margin.

The general blackboard architecture used in music transcription is similar to the one proposed by [Martin \(1996\)](#) (see Fig. 4.13). The blackboard hierarchy is ordered by increasing abstraction, being the input data at the lowest level.

The method from [Bello \(2000\)](#) extends the [Martin \(1996\)](#) architecture by using top-down processing and a neural network to detect the presence or absence of a chord. Other blackboard systems for music transcription have been proposed by [Ellis \(1996\)](#), [Monti and Sandler \(2002\)](#), and [Plumbley et al. \(2002\)](#), and a review on these methods was published by [McKay \(2003\)](#).

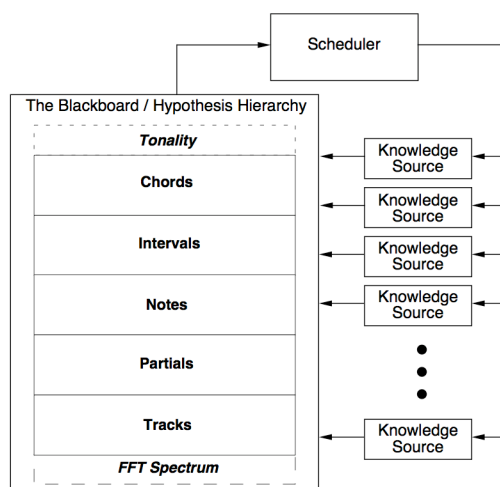


Figure 4.13: Blackboard architecture from [Martin \(1996\)](#).

4.2.10 Database matching

[Bello et al. \(2002\)](#) describes a time-domain approach for piano transcription using a waveforms database. The goal of this method is to use linear algebra to decompose the original signal into a sum of signals that are present in the database. The interesting point is that the waveforms in the database can be known a priori, but they can also be inferred from the analyzed original signal.

To build the adaptive database, the fundamental frequencies are estimated using a simplified version of the blackboard method described in ([Bello, 2000](#)). Once the pitches are estimated, the spectral information of each one is used to resynthesize (using the inverse Fourier transform) an audio signal corresponding to that pitch. To complete the database, covering all the note range, the signals of the missing pitches are calculated from the obtained signals using pitch-shifting by standard phase-vocoder techniques. Once the database is filled, the time-domain method can estimate the pitches.

The method proposed by [Grobler \(2008\)](#) extracts feature vectors from the spectrum data. These vectors are scored against models computed by scaled averages of audio samples of piano and guitar from a training set. The training data consists on sample frames at different note attack levels and distances from the note onset. The predicted pitches are determined using simple distance metrics from the observed feature vector to the dataset feature vectors.

4.3 Discussion of multiple f_0 estimation methods

Multiple f_0 estimation is a complex task, and it is very difficult to obtain a single methodology suitable for all the variety of musical sounds. The advantages and drawbacks among the different categories previously described are discussed in this section.

The input data representation is an important choice. However, time-frequency representations like constant-Q transform, wavelets or filter-banks (see Sec. 2.1.2) do not seem to provide significant advantages over the STFT for this task. As it can be seen in Fig. 2.5, and discussed by Hainsworth (2003) and Yeh (2008), p. 27, multi-resolution schemes do not really solve the time-frequency main issue. Wavelets sacrifice frequency resolution in the high frequencies, which can be a drawback for partial search, and constant-Q and DWT loose temporal precision in the lower frequencies. For these reasons, most of the multiple f_0 estimation methods rely on the STFT.

Most salience methods are computationally efficient, and they provide a mid-level representation which is useful for f_0 estimation. Some of them, are just enhanced representations which need a posterior methodology to estimate the fundamental frequencies. Their main drawback is that in some situations they can lose relevant information or produce spurious components in the signal transformation process. For instance, SACF performs half wave rectification. The FT of a half wave rectified signal (see (Klapuri, 2004), Eq. 4.23) consists of a DC-component, of the original power spectrum scaled down, and of a convolution of the original power spectrum by itself. The convolution of a spectrum by itself produces spectral components at the locations that are multiple of the original spectral intervals, emphasizing partial beating, which can be very useful for single f_0 estimation. However, in polyphonic signals, all the intervals between partials of different sources (and also between different fundamental frequencies) generate prominent components at beating frequencies in the half-wave rectified spectrum, adding spurious data for the posterior analysis.

Iterative cancellation methods are shown to be efficient, but a wrong f_0 estimate may lead to successive errors in an iterative manner and, as pointed out by Yeh (2008), p. 28, these methods can not estimate well partial overlaps or beating, as they do not properly consider source interactions. Matching pursuit methods have similar problems than iterative cancellation methods, as the contribution of each harmonic atom is subtracted at each iteration before processing the residual again.

Joint estimation approaches can handle source interactions better than iterative cancellation methods, but they have high computational costs due to the evaluation of many possible f_0 combinations.

Few supervised learning methods have been used as the core methodology for polyphonic music transcription. Probably, this is because they rely on the data given in the learning stage, and in polyphonic real music the space of observable data is huge. However, supervised learning methods have been successfully applied considering specific instrument transcription (usually, piano) to reduce the search space. The same issue occurs in database matching methods, as they also depend on the ground-truth data.

In contrast to supervised approaches, unsupervised learning methods do not need a priori information about the sources. However, they are suitable for fixed time-frequency profiles (like piano or drums), but modeling harmonic sounds with varying harmonic components remains as a challenge ([Abdallah and Plumbley, 2004](#)).

In general, music transcription methods based on Bayesian models are mathematically complex and they tend to have high computational costs, but they provide an elegant way of modeling the acoustic signal. Statistical spectral model methods are also complex but they are computationally efficient.

Blackboard systems are general architectures and they need to rely on other techniques, like a set of rules ([Martin, 1996](#)) or supervised learning ([Bello, 2000](#)), to estimate the fundamental frequencies. However, the blackboard integration concept provides a promising framework for multiple f_0 estimation.

4.4 Onset detection

In this work, onset detection algorithms have been categorized into two main groups: signal processing methods and machine learning approaches. An extensive review and comparison of different onset detection systems can be found in ([Bello et al., 2005](#)), ([Collins, 2005b](#)), and ([Dixon, 2006](#)).

4.4.1 Signal processing methods

Most onset detection methods are based on signal processing techniques, and they follow the general scheme represented in Fig. 4.14. First, a preprocessing stage is done to transform the signal into a more convenient representation, usually in the frequency domain. Then, an onset detection function (ODF), related to the onset strength at each time frame is defined. A peak picking algorithm is applied to the detection function, and those peaks over a threshold are finally identified as onsets.

In the preprocessing stage, most systems convert the signal to the frequency or complex domain. Besides STFT, a variety of alternative preprocessing methods for onset detection, like filter-banks ([Klapuri, 1999](#)), constant-Q

4. STATE OF THE ART

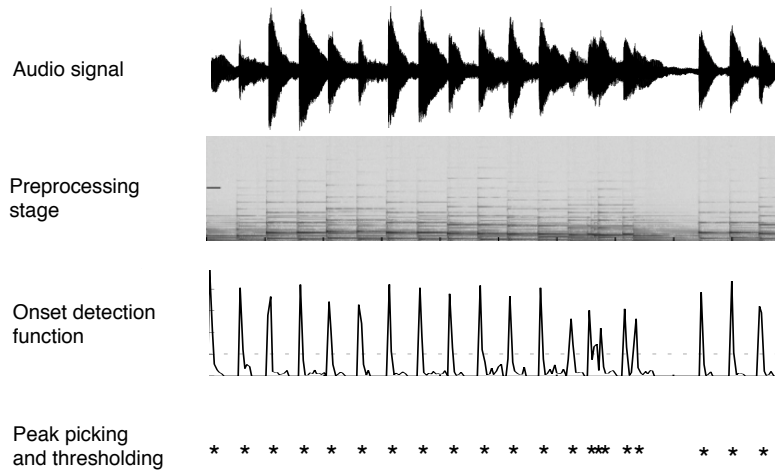


Figure 4.14: General architecture of most onset detection systems.

transform (Duxbury et al., 2002) or wavelet decomposition (Daudet, 2001) can be used.

Some examples of onset detection functions are spectral power, spectral flux, high-frequency content (HFC), phase deviation, weighted phase deviation, complex deviation, and the Kullback-Liebler divergence. Detailed comparatives of different ODFs have been done by Dixon (2006), Bello et al. (2005), and Stowell and Plumbley (2007).

In order to identify the onsets from the ODF, some works just select the peaks over a constant threshold, whereas others use alternative pick picking algorithms like a dynamic median threshold (Brossier et al., 2004, Lee and Kuo, 2006), or a moving average (Lacoste and Eck, 2005).

A categorization of the different onset detection algorithms based on signal processing techniques can be done according to the input data that they use to obtain the ODF.

Energy-based detection

Many onset detection algorithms attempt to detect abrupt energy variations in a time-frequency representation.

The method proposed by Klapuri (1999) is based on a psychoacoustic model. First, in the preprocessing stage, the overall loudness of the signal is normalized, and a bank composed by 21 filters is used to divide the signal into nearly critical-bands. The output of each filter is full-wave rectified and decimated, and amplitude envelopes are extracted for each band. A first order relative difference function is performed for each amplitude envelope, marking the peaks over a

constant threshold as onset components and dropping those components closer than 50 ms. Then, the onset components from separate bands are combined using a loudness model, which assigns a value to each onset candidate, yielding a vector of candidate loudnesses over time. Finally, the peaks of the loudnesses vector that are over a global threshold are accepted as onsets.

Some approaches are based on the scheme proposed by [Klapuri \(1999\)](#), like the algorithm implemented by [Collins \(2005b\)](#) for unpitched sounds. In the preprocessing stage, the STFT is computed, and the spectral bins are combined according to an ERB filter bank. A decibel transformation is done with equal loudness contour correction. Peak picking is performed by scoring local peaks over a seven frame window. A constant threshold is applied to get the onsets. To improve the temporal resolution, the maximum intensities in small blocks are taken in the time domain. An onset time is corrected within the blocks prior to the discovered onset, and this sample position is further corrected to a nearby zero crossing for smooth segmentation.

[Stowell and Plumbley \(2007\)](#) use adaptive whitening as a preprocessing stage, normalizing the magnitude of each spectral bin according to a recent maximum value for that bin, thus obtaining a similar dynamic range over time. Then, different standard ODFs are evaluated and compared using the whitened spectral data. The best results were obtained using the complex deviation function ([Dixon, 2006](#)).

Energy and phase-based detection

Only a few systems like ([Bello and Sandler, 2003](#)) use the phase information alone for onset detection. Most methods that take into account the phase, like ([Bello et al., 2004](#)), also consider the energy variations. In the latter work, abrupt changes of phase and magnitude are combined to get an ODF, and an adaptive median filter is used for peak picking.

[Brossier et al. \(2004\)](#) use a silence gate to discard onsets in quiet regions, and implement a set of onset detection functions based on changes in successive spectral frames, like the high frequency content (HFC) and the complex domain detection function (CDDF) from [Bello et al. \(2004\)](#). Finally, a median dynamic threshold is used, considering as onsets those peaks over the threshold. The system is publicly available as a component of the aubio¹⁸ library.

The method proposed by [Lee et al. \(2007\)](#) decomposes the signal into 6 non-overlapping bands, and applies a forward linear prediction error filter which coefficients are updated with the least mean squares (LMS) algorithm. After getting the linear prediction errors for each band, their envelopes are estimated through rectification, smoothing, and decimation, and they are added together to get a detection function. Besides this detection function, the phase of the

¹⁸Aubio can be downloaded from <http://aubio.org/>

4. STATE OF THE ART

STFT is also considered, obtaining a phase detection function that is combined with the linear prediction estimates to yield the onsets.

Transient-based detection

Although not every transient in a signal correspond to an onset, almost all the musical sounds begin with a transient stage characterized by a non-stationary part and an abrupt amplitude change.

The method proposed by [Röbel \(2005\)](#) performs the STFT to classify the spectral peaks into transient peaks, which are potentially part of an attack, and non transient peaks. This classification is based on the centroid of the time domain energy of the signal segment related to the analyzed peak. A transient statistical model determines whether the spectral peaks identified as transients are produced by background noise or by an onset. The exact onset positions are determined by estimating the starting time of the transient. The value of the detection function is normalized, dividing the transient energy by the total signal energy in the target frame, and a constant threshold is finally applied.

The hybrid approach from [Duxbury et al. \(2002\)](#) aims to detect hard onsets, considering transient energy variations in the upper frequencies, and soft onsets, with a FFT-based distance measure at the low frequencies. To do it, the signal is first split into 5 bands using a constant-Q transform. A transient energy measure is then used to find transient changes in the upper bands, whereas the lowest band is analyzed to yield the standard Euclidean distance between two consecutive FFT vectors. The detection function is based on the difference between the signal for each band and a smoothed version of itself. The onsets are detected using an automatic threshold based on a mixture of Gaussians or, alternatively, on the derivative of the onsets histogram. Finally, onsets across bands are combined to yield the final estimate through a weighted scheme.

Energy and pitch-based detection

Some methods combine energy information with pitch-related features.

Besides using the RTFI method ([Zhou and Mattavelli, 2007](#)) for multiple f_0 estimation, this time-frequency processing technique was also applied by [Zhou et al. \(2008\)](#) for onset detection. In this method, the RTFI spectrum is transformed through signal processing techniques into a difference pitch energy spectrum and a normal pitch energy spectrum. The onset is classified according to the difference pitch energy spectrum into hard or soft onset. If the onset is marked as hard, then an energy-based algorithm is selected. This algorithm uses the difference pitch energy spectrum across all frequency channels to generate the detection function, which is smoothed by a moving average filter to find the onsets with a simple peak picking operation. If the onset type is soft, then a

pitch-based detection algorithm is selected instead. The pitch-based algorithm uses the normal pitch energy spectrum to divide the signal into transient and stable parts, and the transient part is used to locate the onsets by inspecting the energy changes.

4.4.2 Machine learning methods

Some methods do not follow the general scheme of Fig. 4.14. Instead, they use machine learning techniques to classify each frame into onset or non-onset.

Supervised learning

The system proposed by [Lacoste and Eck \(2007\)](#) use either STFT and constant Q transforms in the preprocessing stage. The linear and logarithmic frequency bins are combined with the phase plane to get the input features for one or several feed-forward neural networks, which classify frames into onset or non onset. In the multiple network architecture, the tempo trace is also estimated and used to condition the probability for each onset. This tempo trace is computed using the cross-correlation of the onset trace with the onset trace autocorrelation within a temporal window. A confidence measure that weights the relative influence of the tempo trace is provided to the network.

[Marolt et al. \(2002\)](#) use a bank of 22 auditory filters to feed a fully connected network of integrate-and-fire neurons. This network outputs a series of impulses produced by energy oscillations, indicating the presence of onsets in the input signal. Due to noise and beating, not all the impulses correspond to onsets. To decide which impulses are real onsets, a multilayer perceptron trained with synthesized and real piano recordings is used to yield the final estimates.

Support vector machines have also been used for onset detection by [Kapanci and Pfeffer \(2004\)](#) and [Davy and Godsill \(2002\)](#) to detect abrupt spectral changes.

Unsupervised learning

Unsupervised learning techniques like NMF and ICA have been also applied for onset detection.

[Wang et al. \(2008\)](#) generate the non-negative matrices with the magnitude spectra of the input data. The basis matrices are the temporal and frequency patterns. The temporal patterns are used to obtain three alternative detection functions: a first-order difference function, a psychoacoustically motivated relative difference function, and a constant-balanced relative difference function. These ODFs are similarly computed by inspecting the differences of the temporal patterns.

4. STATE OF THE ART

[Abdallah and Plumbley \(2003b\)](#) consider onsets as surprising moments in the waveform. The detection is based on a probability model of the input, which generates a moment-by-moment trace of the probability of each observation. In this method, ICA is used as a conditional probability model, and the probability assigned by the model to each observation is also used as a form of data for further analysis.

4.5 Discussion of onset detection methods

Like in multiple f_0 estimation, finding a general purpose onset detection algorithm is a challenging task due to the different onset characteristics. For instance, smooth onsets need longer temporal information than hard (percussive) onsets. Citing [Tan et al. \(2009\)](#), due to the inherent variable nature of onsets resulting from the different types of instruments, no simple algorithm will be optimal for general onset detection.

Transient and energy are features related to hard onsets, whereas phase and pitch can be used to detect soft onsets produced by some harmonic sounds. In onset detection, a good trade-off is hard to achieve, and the combined approaches considering both hard and soft onsets, i.e. using energy and pitch information, or energy and phase, are probably the most versatile methods for this task.

5

Onset detection using a harmonic filter bank

A novel approach for onset detection is presented in this chapter. The audio signal is analyzed through a 1/12 octave (one semitone) band-pass filter bank simulated in the frequency domain, and the temporal derivatives of the filtered values are used to detect spectral variations related to note onsets.

As previously described, many onset detection methods apply a preprocessing stage by decomposing the signal into multiple frequency bands. In the perceptually motivated onset detector proposed by Klapuri (1999), a set of critical band filters is used. Scheirer (1998) uses a six band filter bank, each one covering roughly one octave range, and Duxbury et al. (2002) performs a sub-band decomposition of the signal.

The motivation of the proposed approach is based on the characteristics of most harmonic pitched sounds. The first 5 harmonics of a tuned sound coincide¹ with the frequencies of other pitches in the equal temperament (see Fig. 2.16). Other characteristic of these sounds is that usually most of their energy is concentrated in the first harmonics. A one semitone filter bank is composed by a set of triangular filters, which center frequencies coincide with the musical pitches (see Fig. 5.1).

In the sustain and release stages of a sound, there can be slight variations in the intensity (tremolo) and the frequency (vibrato) of the harmonics. For instance, a harmonic peak at the frequency bin k in a given frame can be shifted to the position $k + 1$ in the following frame. In this scenario, direct spectra comparison, like spectral flux (see Eq. 2.14), may yield false positives, as intensity differences are detected. Using this musically motivated filter bank, the value of the band which center is close to k will be similar in both frames, avoiding a false detection.

¹with slight deviations when inharmonicity is present.

5. ONSET DETECTION

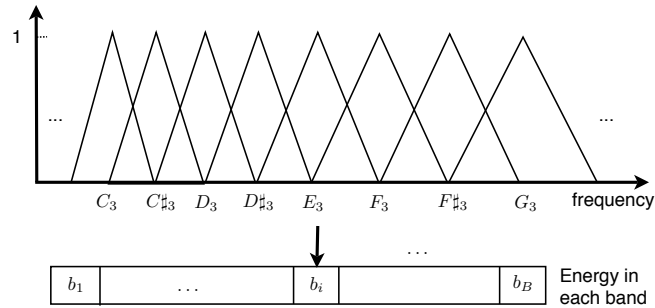


Figure 5.1: One semitone filter bank.

Therefore, by using one semitone filters, the effects of subtle spectrum variations produced during the sustain and release stages of a note are minimized, whereas in the attack the filtered amplitudes increase significantly, as most of the energy of the partials is concentrated in the center frequencies of the semitone bands. This way, the system is specially sensitive to frequency variations that are larger than one semitone. Therefore, the method is primarily based on energy changes, but also considering the harmonic properties of the sounds.

The proposed approach has been evaluated and compared to other works. Some contents of this chapter have been published in (Pertusa et al., 2005) and (Pertusa and Iñesta, 2009). The algorithm, developed in C++, has been publicly released² for research purposes.

5.1 Methodology

For detecting the beginnings of the notes in a musical signal, the method analyzes the spectrum information across one semitone filter bank, computing the band differences in time to obtain a detection function. Peaks in this function are extracted, and those which values are over a threshold are considered as onsets.

5.1.1 Preprocessing

From a digital audio signal, the STFT is computed, providing its magnitude spectrogram. A Hanning window with 92.9 ms length is used, with a 46.4 ms hop size. With these values, the temporal resolution achieved is $\Delta t = 46.4$ ms, and the spectral resolution is $\Delta f = 10.77$ Hz.

²http://grfia.dlsi.ua.es/cm/worklines/pertusa/onset/pertusa_onset.tgz

Using a 1/12 octave filter bank, the filter corresponding to the pitch $G\sharp_0$ has a center frequency of 51.91 Hz, and the fundamental frequency of the next pitch, A_0 , is 55.00 Hz, therefore this spectral resolution is not enough to build the lower filters. Zero padding was used to get more points in the spectrum. Using a zero padding factor $z = 4$, three additional windows with all samples set to zero were appended at the end of each frame before doing the STFT. With this technique, a frequency resolution $\Delta f = 10.77/4 = 2.69$ Hz is eventually obtained.

At each frame, the spectrum is apportioned among a one semitone filter bank to produce the corresponding filtered values. The filter bank comprises from 52 Hz (pitch $G\sharp_0$) to the Nyquist frequency to cover all the harmonic range. When $f_s = 22,050$ Hz, $B = 94$ filters are used³, which center frequencies correspond to the fundamental frequencies of the 94 notes in that range. The filtered output at each frame is a vector \mathbf{b} with B elements ($\mathbf{b} \in \mathbb{R}^B$).

$$\mathbf{b} = \{b_1, b_2, \dots, b_i, \dots, b_B\} \quad (5.1)$$

Each value b_i is obtained from the frequency response H_i of the corresponding filter i with the spectrum. The Eq. 5.2 is used⁴ to compute the filtered values:

$$b_i = \sqrt{\sum_{k=0}^{K-1} (|X[k]| \cdot |H_i[k]|)^2} \quad (5.2)$$

5.1.2 Onset detection functions

Two onset detection functions have been used. The first one, called $o[t]$, can be used for percussive (hard) onsets, whereas an alternative ODF $\tilde{o}[t]$ has been proposed for sounds with smooth (soft) onsets.

Onset detection function for hard onsets ($o[t]$)

Like in other onset detection methods, as (Bilmes, 1993), (Goto and Muraoka, 1995, 1996), and (Scheirer, 1998), a first order derivative function is used to pick potential onset candidates. In the proposed approach, the derivative $c[t]$ is computed for each filter i .

$$c_i[t] = \frac{d}{dt} b_i[t] \quad (5.3)$$

³When $f_s = 44,100$ Hz, there are $B = 106$ bands.

⁴This equation was selected instead of Eq. 2.9 since it experimentally yielded better results for this task.

5. ONSET DETECTION

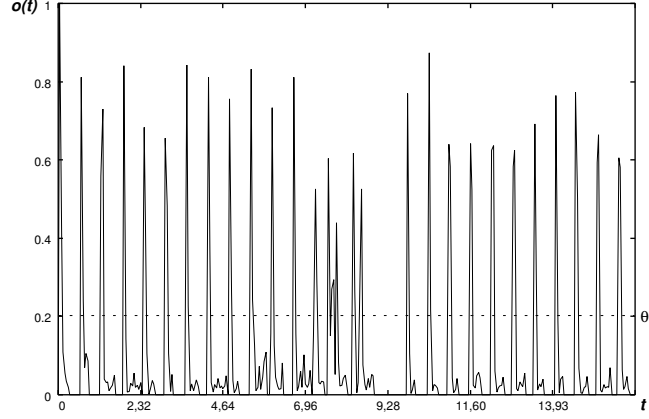


Figure 5.2: Example of the onset detection function $o[t]$ for a piano melody, RWC-MDB-C-2001 No. 27 from Goto (2003), RWC database.

The values for each filter must be combined to yield the onsets. In order to detect only the beginnings of the events, the positive first order derivatives of all the bands are summed at each time, whereas negative derivatives, which can be associated with offsets, are discarded:

$$a[t] = \sum_{i=1}^B \max\{0, c_i[t]\}. \quad (5.4)$$

To normalize the onset detection function, the overall energy $s[t]$ is also computed (note that $a[t] < s[t]$):

$$s[t] = \sum_{i=1}^B b_i[t] \quad (5.5)$$

The sum of the positive derivatives $a[t]$ is divided by the sum of the filtered values $s[t]$ to compute a relative difference. Therefore, the onset detection function $o[t] \in [0, 1]$ is:

$$o[t] = \frac{a[t]}{s[t]} = \frac{\sum_{i=1}^B \max\{0, c_i[t]\}}{\sum_{i=1}^B b_i[t]} \quad (5.6)$$

Fig. 5.2 shows an example of the onset detection function $o[t]$ for a piano excerpt, where all the peaks over the threshold θ were correctly detected onsets.

Onset detection function for soft onsets ($\tilde{o}[t]$)

The previous methodology yields good results for instruments that have a sharp attack, like a piano or a guitar. But for instruments with a very smooth attack, like violins, more frames should be considered. For these sounds, Eq. 5.3 can be replaced by:

$$\tilde{c}_i[t] = \sum_{j=1}^C j \cdot (b_i[t+j] - b_i[t-j]) , \quad (5.7)$$

being C the number of adjacent frames considered. This expression is based on the method proposed by Young et al. (2000), Eq. 5.16, to enhance the performance of a speech recognition system.

Using this scheme, the difference is centered on each particular frame, thus two side difference (with $C = 1$) is used instead of the frame itself. When using $C = 2$, the derivative is calculated for a longer period. Using Eq. 5.7,

$$\tilde{a}[t] = \sum_{i=1}^B \max \{0, \tilde{c}_i[t]\} \quad (5.8)$$

With these equations, Eq. 5.5 must be replaced by Eq. 5.9 to normalize $\tilde{o}[t]$ into the range $[0, 1]$:

$$\tilde{s}[t] = \sum_{i=1}^B \sum_{j=1}^C j \cdot b_i[t+j] \quad (5.9)$$

Therefore, $\tilde{o}[t]$ is calculated as:

$$\tilde{o}[t] = \frac{\tilde{a}[t]}{\tilde{s}[t]} = \frac{\sum_{i=1}^B \max \{0, \tilde{c}_i[t]\}}{\sum_{i=1}^B \sum_{j=1}^C j \cdot b_i[t+j]} \quad (5.10)$$

An example of the onset detection function for a violin sound is shown in Fig. 5.3, without considering additional frames (a), with $C = 1$ (b), and with $C = 2$ (c).

Note that the higher C is, the lower the temporal precision is for detecting onsets, but the success rate can improve in some instruments which onsets are difficult to detect. For an adequate detection, the notes must have a duration $l \geq \Delta t(C + 1)$. With the used parameters, if $C = 2$ then $l = 139.2$ ms, so this methodology is not suitable for very rapid onsets⁵.

⁵139 ms is the duration of a sixteenth note when tempo = 107 bpm.

5. ONSET DETECTION

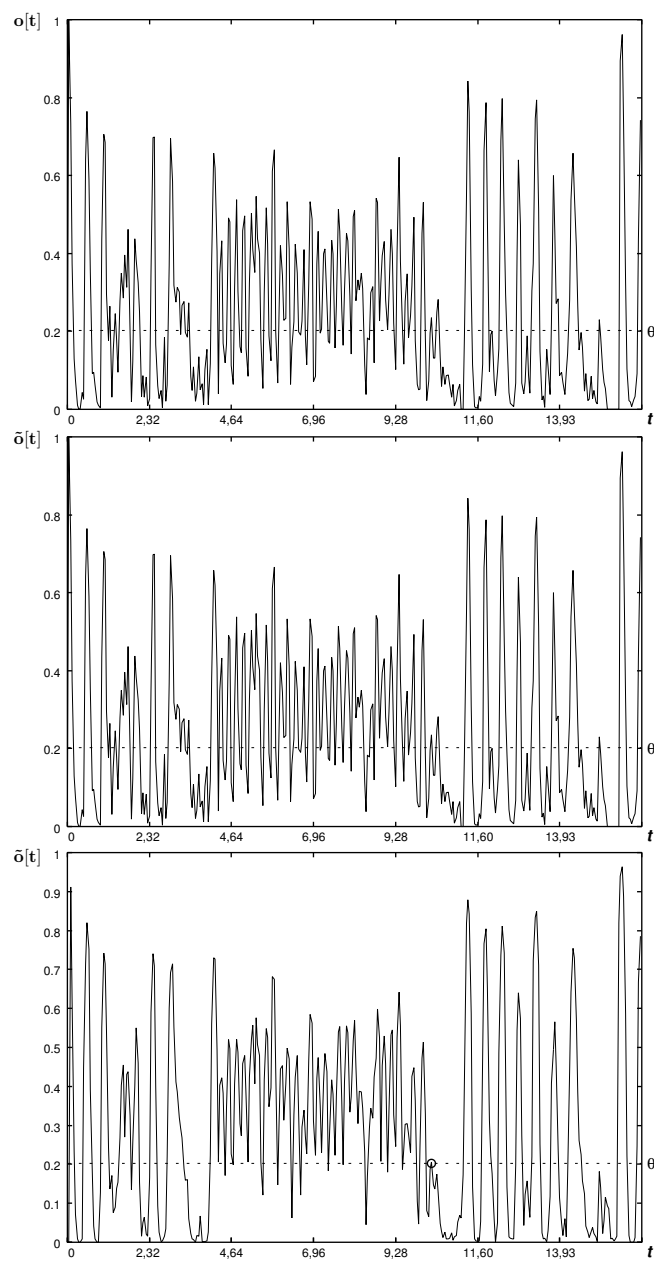


Figure 5.3: Onset detection function for a polyphonic violin song (RWC-MDB-C-2001 No. 36 from Goto (2003), RWC database). (a) $o[t]$; (b) $\tilde{o}[t]$, with $C = 1$; (c) $\tilde{o}[t]$, with $C = 2$. With $C = 2$, all the onsets were successfully detected except by one, which is marked with a circle.

5.1.3 Peak detection and thresholding

The last stage is to extract the onsets from the onset detection function. Peaks at time t are identified in the onset detection function when $o[t-1] < o[t] > o[t+1]$, and those peaks over a fixed threshold $o[t] > \theta$ are considered as onsets. Two consecutive peaks can not be detected, therefore the minimum temporal distance between two onsets is $2\Delta t = 92.8$ ms. A silence threshold μ is also introduced to avoid false positive onsets in quiet regions, in such a way that if $s[t] < \mu$, then $o[t] = 0$. The same peak detection and thresholding procedure is applied for $\tilde{o}[t]$.

The silence gate μ is only activated when silences occur, or when the considered frame contains very low energy, therefore it is not a critical parameter. The precision/recall deviation can be controlled through the threshold θ .

5.2 Evaluation with the ODB database

To evaluate the method and set up the thresholds, a test database called ODB (onset detection database) has been built using a set of real recordings⁶. The ground-truth onset positions were marked and reviewed (see Fig. 5.4) using the *speech filling system* (SFS⁷) software. The ODB data set contains a number of sounds selected from the Goto (2003) RWC database, plus other real recordings. The songs were selected to cover a relatively wide range of instruments and musical genres.

The ODB database, with the audio signals and the labeled onsets, has been publicly released⁸ for research purposes. An onset detection evaluation software, which is also available⁹ has been developed to compare the detected onsets with the ground-truth. This algorithm computes the number of correct detections, false positives, and false negatives, considering a 50 ms error margin.

5.2.1 Results using $o[t]$

As previously described, the system has two free parameters; the silence gate threshold μ , to avoid false positives when the signal level is very low, and the onset detection threshold θ , which controls the precision/recall deviation of the ODF. The method was evaluated with the ODB database to set an appropriate value for θ . The results are shown in Fig. 5.5. A good compromise between precision and recall was obtained using $\theta = 0.18$, with $\mu = 70$.

⁶Thanks to Jason Box for labeling this data set.

⁷<http://www.phon.ucl.ac.uk/resource/sfs/>

⁸<http://grfia.dlsi.ua.es/cm/worklines/pertusa/onset/ODB>

⁹<http://grfia.dlsi.ua.es/cm/worklines/pertusa/onset/evaluator>

5. ONSET DETECTION

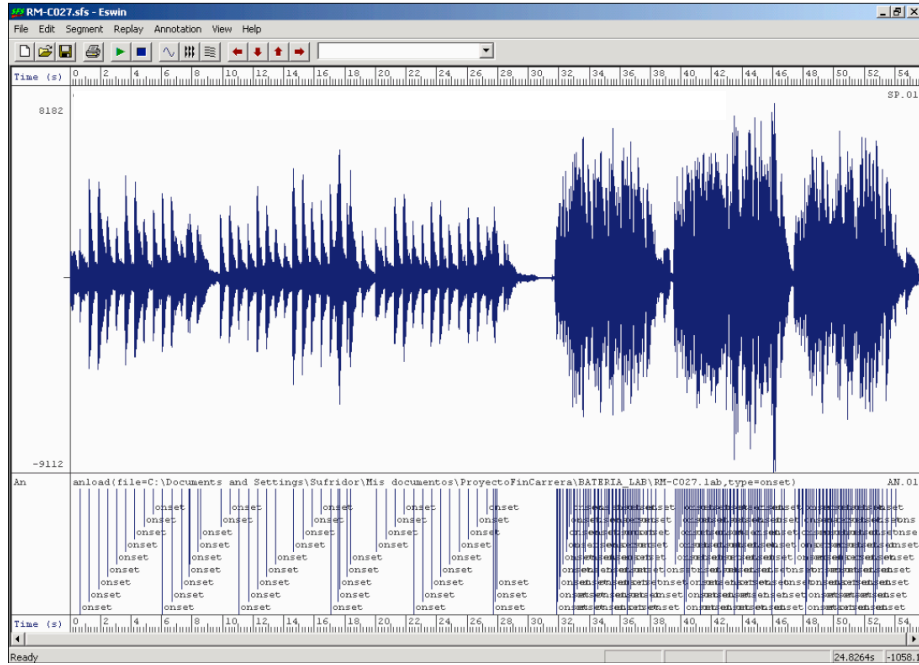


Figure 5.4: Onsets from RWC-MDB-C-2001 No. 27 from Goto (2003), RWC database, labeled with speech filling system (SFS).

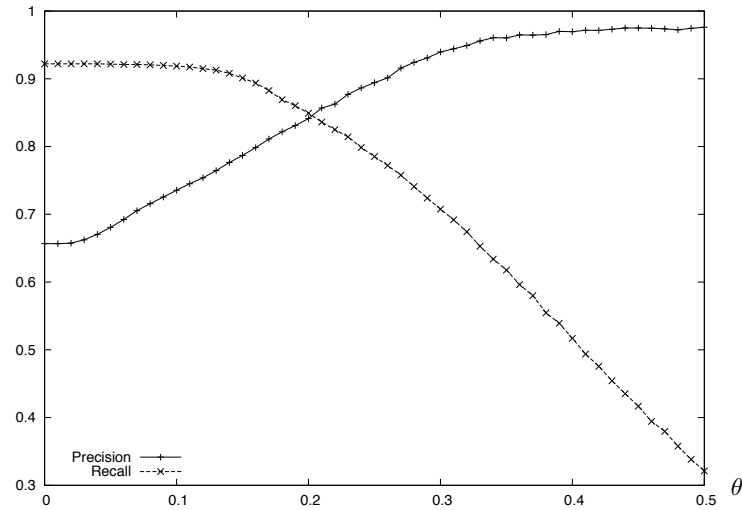


Figure 5.5: Onset detection ($o[t]$) precision and recall curves in function of the threshold θ , using a constant value for the silence threshold $\mu = 70$.

5.2. EVALUATION WITH THE ODB DATABASE

Reference	Content	OK	FP	FN	M	D	Pr %	Re %	F-m %
RWC-C02	classic	64	37	43	2	2	66.37	59.81	61.54
RWC-C03	classic	25	41	31	0	0	37.88	44.64	40.98
RWC-C26	piano	36	1	0	0	0	97.30	100.0	98.63
RWC-C27	piano	210	0	13	0	0	100.0	94.17	97.00
RWC-C36	violin	45	19	0	0	0	70.31	100.0	82.57
RWC-C38	violin	165	13	29	0	0	92.70	85.05	88.71
RWC-J01	piano	96	6	17	0	0	94.12	84.96	89.30
RWC-G08	rock	62	17	8	0	0	78.48	88.57	83.22
2-artificial	soul	117	35	16	1	1	76.97	87.97	82.11
2-uncle_mean	jazz	157	10	21	0	0	94.01	88.20	91.01
3-long_gone	rock	135	53	8	0	0	71.81	94.41	81.57
3-you_think_too_much	jazz	160	10	25	0	0	94.12	86.49	90.14
6-three	rock	138	27	17	0	0	83.64	89.03	86.25
8-ambriellb	electro	111	33	36	0	0	77.08	75.51	76.29
15-tamerlano	opera	127	41	12	0	0	75.60	91.37	82.74
25-rujero	guitar	92	17	2	0	0	84.40	97.87	90.64
Its_alright_for_you	rock	83	1	1	0	0	98.81	98.81	98.81
Tiersen 11	bells	37	36	1	0	0	50.68	97.37	66.67
Realorgan3	organ	13	10	2	0	0	56.52	86.67	68.42
Total		1873	406	282	3	3	82.19	86.91	84.48

Table 5.1: Onset detection results using the proposed database (ODB). The table shows the number of correctly detected onsets (OK), false positives (FP), false negatives (FN), merged onsets (M), doubled onsets (D), precision (P), recall (R), and F-measure (F-m).

The detailed results using $o[t]$ with these thresholds can be seen in Tab. 5.1. The overall F-measure achieved was 84.48%.

In order to get a perceptual evaluation of the results, once performed the onset detection, new audio files¹⁰ were generated using CSound by adding to the original waveform a click sound in the positions where the onsets were detected.

Comparison with other approaches

In order to compare the method with other approaches, two publicly available onset detection algorithms were evaluated using the ODB database. The experiments were done comparing the onset times obtained by BeatRoot¹¹ and aubio¹² with the ground-truth onsets of the ODB database using the evaluation methodology previously described.

BeatRoot, introduced by Dixon (2006), is a software package for beat tracking, tempo estimation and onset detection. To evaluate the method, the onset times were obtained using the BeatRoot-0.5.6 default parameters with the following command:

```
java -jar beatroot-0.5.6.jar -o onsets.txt -0 input.wav
```

¹⁰http://grfia.dlsi.ua.es/cm/worklines/pertusa/onset/generated_sounds_ODB

¹¹<http://www.elec.qmul.ac.uk/people/simond/beatroot/index.html>

¹²<http://aubio.org/>

5. ONSET DETECTION

System	OK	FP	FN	M	D	Pr %	Re %	F-m %
Pertusa et al. (2005)	1873	406	282	3	3	82.19	86.91	84.48
Brossier (2005) - aubio	1828	608	327	79	80	75.04	84.83	79.63
Dixon (2006) - BeatRoot	1526	778	629	21	21	66.23	70.81	68.45

Table 5.2: Comparison with other methods using the ODB database and $o[t]$.

In the method from [Dixon \(2006\)](#), different onset detection functions based on spectral flux, phase deviation, and complex domain¹³ can be selected. The onset detection function values are normalized and a simple peak picking algorithm is used to get the onset times.

`Aubio` is the implementation of the algorithm proposed by [Brossier \(2005\)](#), submitted to the MIREX 2005 contest and previously described in Sec. 4.4. Like in `BeatRoot`, the default parameters were used for the evaluation of this method:

```
aubioonset -i input.wav > onsets.txt
```

The results (Tab. 5.2) show that the proposed method outperforms these two approaches using the ODB data set.

5.2.2 Results using $\tilde{o}[t]$

For pitched instruments with a non-percussive onset, like a violin or a church organ, more frames should be considered due to the longer attack stage. In the ODB test set, these kind of sounds only appear in five songs. The evaluation results for these melodies using $C = 2$ and $\theta = 0.18$ are shown in Tab. 5.3. The results with $C = 1$ were quite similar to those obtained without considering additional frames, therefore they are not shown in the table.

In the songs missing in the table, i.e., those with hard onsets, the F-measure importantly decreases as the number of frames increases, mainly due to the worse temporal resolution. Using $\tilde{o}[t]$ for the non-percussive onset audio files, the system only yielded better results for the RWC-C03 and RWC-C36 melodies, as shown in Tab. 5.3.

Therefore, this methodology is only suitable for very specific musical pieces, and it is not adequate for a general purpose onset detection method, yielding better results only in a very small subset of non-percussive pitched sounds.

¹³Based on the estimation of the expected amplitude and phase of the current bin according to the previous two bins.

Reference	OK	FP	FN	M	D	Pr %	Re %	F-m %
RWC-C02	47	35	60	0	0	57.32	43.93	49.74
RWC-C03	25	38	31	0	0	39.68	44.64	42.02
RWC-C36	45	5	0	0	0	90.00	100.00	94.74
RWC-C38	101	45	93	5	5	69.18	52.06	59.41
Realorgan3	11	13	4	0	0	45.83	73.33	56.41

Table 5.3: Results using $\tilde{o}[t]$, with $C = 2$.

5.3 MIREX evaluation

In order to compare the proposed method with other approaches using an independent and large data set, it was evaluated in the [MIREX \(2005\)](#) onset detection contest. As previously discussed, $\tilde{o}[t]$ is not suitable for general purpose onset detection and the results were not satisfactory for most sounds, therefore the algorithm was submitted only with the $o[t]$ function.

For this challenge, the method was implemented using D2K and M2K modules. The D2K (data to knowledge) toolkit¹⁴ is a modular environment for data mining, whereas M2K¹⁵ is a D2K module set for music information retrieval applications developed by the IMIRSEL¹⁶ project.

The D2K itinerary submitted to [MIREX \(2005\)](#) is shown in [Fig. 5.6](#). Unfortunately, there was a bug in the code which caused a very low F-measure in the evaluation. The problem was that the mean deviation from the ground-truth onsets was of -22 ms, because all the onset times were reported as the beginning time of the frame, instead of the center.

As the method was not properly evaluated in [MIREX \(2005\)](#), the algorithm was implemented again in C++ and resubmitted to the [MIREX \(2009\)](#) onset evaluation contest. The [MIREX \(2009\)](#) data set and the evaluation metrics are exactly the same used in previous contests ([MIREX, 2005, 2006, 2007](#)), therefore the results are comparable with those obtained in previous editions.

5.3.1 Methods submitted to MIREX 2009

Besides the proposed method, three algorithms were evaluated in the [MIREX \(2009\)](#) onset detection contest. These algorithms are briefly described next.

The onset detection function from [Tzanetakis \(2009\)](#) is based on the half-wave rectified spectral flux. It uses a pick picking algorithm to find local maxima in consecutive frames and a threshold relative to the local mean. To reduce the

¹⁴<http://alg.ncsa.uiuc.edu/do/tools/d2k>

¹⁵Music to knowledge, <http://www.music-ir.org/evaluation/m2k/>

¹⁶International Music Information Retrieval Systems Evaluation Laboratory.

5. ONSET DETECTION

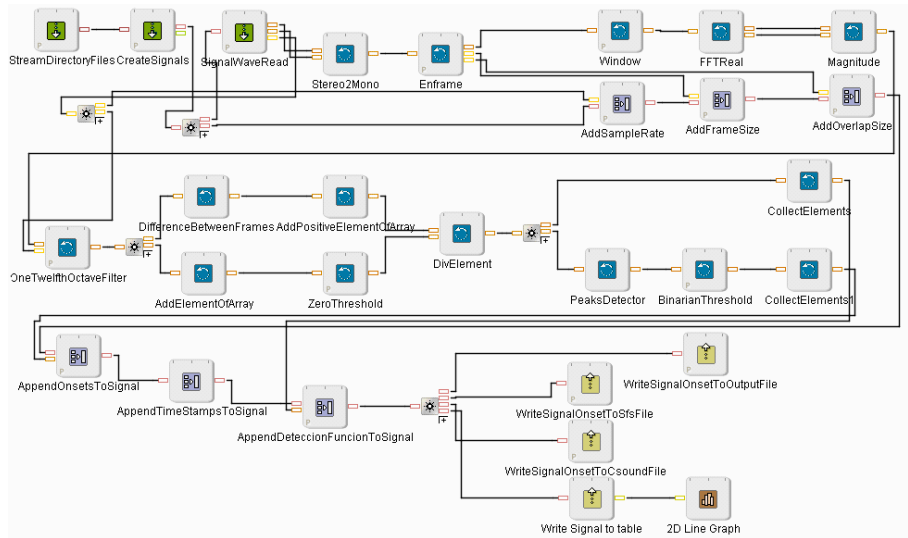


Figure 5.6: D2K itinerary of the system submitted to [MIREX \(2005\)](#).

number of false positives, the ODF is smoothed using a Butterworth filter¹⁷ both in the forward and reverse directions to avoid phase distortions.

The [Röbel \(2009\)](#) method is based on the classification of spectral peaks into transients and non-transients. The algorithm is very similar to that of [Röbel \(2005\)](#)¹⁸, as it analyzes the transient peak probability in the different spectral bands. The main difference from the previous approach is the use of bands with harmonically related center frequencies, instead of continuous frequency regions, to better detect pitched transients. Five versions were submitted to the [MIREX \(2009\)](#) onset detection contest. The algorithms labeled as `12_nhd` and `16_nhd` were trained with the same data set used in ([Röbel, 2005](#)), whereas the algorithms `7_hd`, `10_hd` and `19_hdc` were trained with additional synthesized sounds.

The method proposed by [Tan et al. \(2009\)](#) integrates energy-based and pitch-based detections. First, the system attempts to distinguish between pitched percussive onsets (like pianos), non-percussive onsets (like brasses or winds), and unpitched onsets (like drums). This is done by detecting the average bandwidth of abrupt energy changes (percussiveness) and estimating the pitch tuning in a similar way than in ([Zhu and Kankanhalli, 2006](#)). A musical piece is classified into one of these three categories using heuristic rules over the percussiveness and the pitch tuning measures.

¹⁷A [Butterworth \(1930\)](#) filter is designed to have a frequency response which is maximally flat in the passband.

¹⁸See Pag. 80 for a description of this method.

Participant	OK	FP	FN	M	D	Pr %	Re %	F-m %
Röbel (2009) 10_hd	7015	1231	2340	161	133	85.00	79.19	79.60
Röbel (2009) 7_hd	7560	2736	1795	188	257	81.32	83.30	79.00
Röbel (2009) 19_hdc	7339	2367	2016	185	212	80.56	81.88	78.31
Pertusa and Iñesta (2009)	6861	2232	2494	196	10	79.99	77.50	76.79
Röbel (2009) 16_nhd	6426	846	2929	148	183	86.39	73.62	76.48
Röbel (2009) 12_nhd	6440	901	2915	145	198	85.96	73.15	76.10
Tan et al. (2009) 1	6882	2224	2473	157	308	75.67	76.97	74.43
Tan et al. (2009) 2	6588	1976	2767	152	266	78.28	74.58	73.38
Tan et al. (2009) 3	5961	1703	3394	146	285	79.61	68.97	68.63
Tan et al. (2009) 5	7816	5502	1539	84	1540	62.88	83.69	68.23
Tan et al. (2009) 4	5953	1843	3402	135	345	78.98	68.91	67.94
Tzanetakis (2009)	5053	2836	4302	162	46	67.01	59.91	59.54

Table 5.4: Overall MIREX 2009 onset detection results ordered by F-measure. The precision, recall and F-measure are averaged. The highest F-measure was obtained using $\theta = 0.25$.

If the musical excerpt is classified as unpitched, the onset detection is based only on energy processing. If it is set as pitched (percussive or non-percussive), both energy processing and pitch processing are combined. The energy-based processing computes the spectral differences from two consecutive frames and applies an adaptive threshold. The pitch-based processing computes the chromagram and looks for changes in the strongest base pitch class and dominant harmonics pitch class pair. Adjacent time frames with the same pitch content are grouped into the same cluster, and the clusters indicate regions belonging to the same note. In the percussive pitched class, energy changes have higher weights than pitch changes, whereas in the non-percussive category the pitch changes are considered to be more relevant.

5.3.2 MIREX 2009 onset detection results

The method was evaluated in the [MIREX \(2009\)](#) onset detection contest using different values of $\theta \in [0.1, 0.3]$. The overall results using the best parameters for each algorithm are shown in Tab. 5.4. The proposed method yielded a good average F-measure with a very low computational cost (see Tab. 5.5).

The Tab. 5.6 shows the values of θ that yielded the best results for each sound category compared with the highest overall average F-measure among the evaluated methods. The proposed approach achieved the highest average F-measure for the brass, drums and plucked strings categories, characterized by hard onsets.

Complex sounds are mixtures of unpitched and pitched sounds, including singing voice. In this category, the method achieved a lower F-measure than using poly-pitched and drum sounds probably due to the presence of singing voice. In general, this algorithm is not suitable for singing voice, which is usually not perfectly tuned and tends to have partials shifting in frequency across different semitones, causing many false positives.

5. ONSET DETECTION

Participant	Runtime (hh:mm)
Pertusa and Iñesta (2009)	00:01
Tzanetakis (2009)	00:01
Röbel (2009) 12_nhd	00:02
Röbel (2009) 16_nhd	00:02
Röbel (2009) 7_hd	00:03
Röbel (2009) 19_hdc	00:03
Röbel (2009) 10_hd	00:04
Tan et al. (2009) 1	01:57
Tan et al. (2009) 2	01:57
Tan et al. (2009) 3	01:57
Tan et al. (2009) 4	01:57
Tan et al. (2009) 5	01:57

Table 5.5: MIREX 2009 onset detection runtimes.

Class	Files	θ	Pr %	Re %	F-m %	Best F-m %
Complex	15	0.19	68.97	70.25	68.51	74.82
Poly-pitched	10	0.20	93.58	88.27	90.36	91.56
Solo bars & bells	4	0.26	79.69	80.12	77.34	99.42
Solo brass	2	0.24	77.69	83.68	79.88	79.88
Solo drum	30	0.21	94.00	88.93	90.68	90.68
Solo plucked strings	9	0.28	91.25	89.56	90.11	90.11
Solo singing voice	5	0.30	15.17	46.12	22.63	51.17
Solo sustained strings	6	0.24	58.06	60.92	55.47	64.01
Solo winds	4	0.28	54.97	70.17	60.15	75.33

Table 5.6: Detailed MIREX 2009 onset detection results for the proposed method with the best θ for each class. The precision, recall and F-measure are averaged. The best F-measure among the evaluated methods is also shown.

Participant	Params	Pr %	Re %	F-m %
Röbel (2009) 19_hdc	0.34	92.07	92.02	91.56
Röbel (2009) 7_hd	0.34	91.70	92.31	91.54
Pertusa and Iñesta (2009)	0.20	93.58	88.27	90.36
Röbel (2009) 19_hd	0.52	93.02	87.79	89.20
Röbel (2009) 16_nhd	0.43	98.51	80.21	87.26
Röbel (2009) 12_nhd	0.49	96.12	80.97	87.11
Tan et al. (2009) 1	N/A	85.94	83.11	83.12
Tan et al. (2009) 2	N/A	85.94	83.11	83.12
Tan et al. (2009) 3	N/A	89.61	70.73	74.17
Tan et al. (2009) 4	N/A	89.23	70.32	73.77
Tan et al. (2009) 5	N/A	61.33	90.01	68.66
Tzanetakis (2009)	N/A	71.91	66.43	67.41

Table 5.7: MIREX 2009 poly-pitched results ordered by average F-measure.

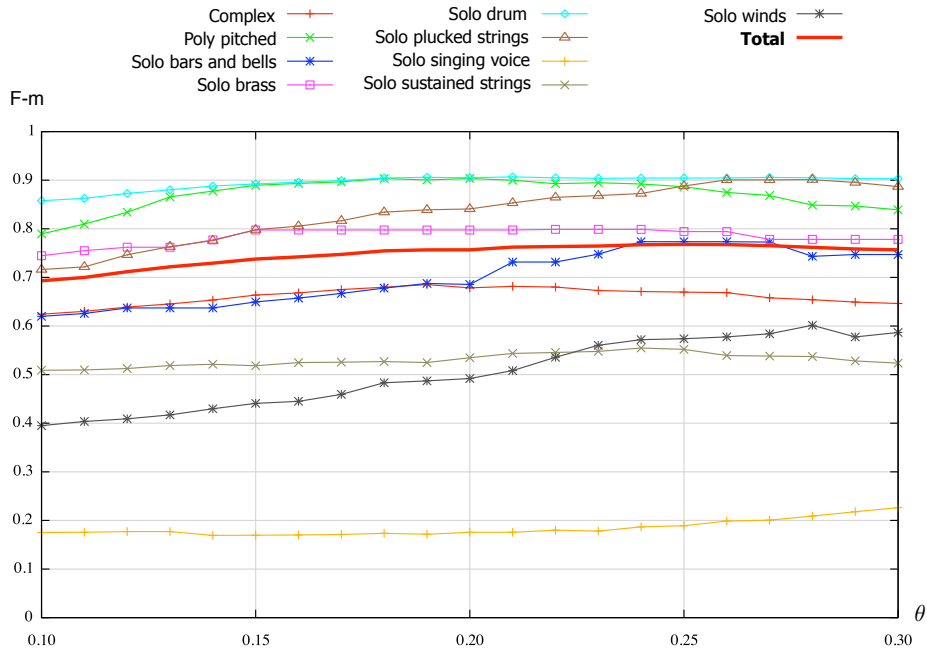


Figure 5.7: MIREX 2009 onset detection F-measure respect to the threshold θ for the different sound classes using the proposed method.

The proposed methodology is primarily intended for detecting pitch changes, therefore the poly-pitched results (see Tab. 5.7) are of special interest. For this class of sounds, the F-measure was close to the best. These results are satisfactory, given that the R obel (2009) and Tan et al. (2009) approaches are also oriented to the onset detection of pitched sounds.

As expected, the method performs slightly worse with sounds with non-percussive attacks, like sustained strings and winds. For instance, for sax sounds two onsets are usually detected by the system; one when the transient (breathing) begins, and other when the pitch is reached. Portamentos¹⁹ are also a problem for the proposed method, and they usually occur in these kind of sounds, and also in singing voice. A strong portamento produces that a new onset is detected each time that a semitone is reached, like it would happen with a glissando. This is not a drawback for multiple pitch estimation systems, but it may yield some false positive onsets. Therefore, for detecting the onsets of these sounds, it is probably more adequate to identify the transients rather than the pitch changes.

¹⁹A portamento is a continuous and smooth frequency slide between two pitches. A glissando is a portamento which moves in discrete steps corresponding to pitches. For instance, a glissando can be played with a piano, but this instrument is unable to play a portamento. Violins can produce portamentos, although they can also generate glissandos.

5. ONSET DETECTION

Bars and bells have percussive onsets and they are typically pitched, although most of these sounds are inharmonic. Therefore, their energy may not be concentrated in the central frequencies of the one semitone bands. In the proposed method, when this happens and the harmonics slightly oscillate in frequency, they can easily reach adjacent bands, causing some false positives. Anyway, it is difficult to derive conclusions for this class of sounds, as only 4 files were used for the evaluation and the MIREX data sets are not publicly available.

Interestingly, the proposed approach also yielded good results with unpitched sounds, and it obtained the highest F-measure in solo-drum excerpts among all the evaluated methods.

The best threshold value for poly-pitched and complex sounds was around $\theta = 0.20$, which coincides with the threshold experimentally obtained with the ODB database. Using this threshold, the overall F-measure is only 1% lower (see Fig. 5.7) than with the best threshold $\theta = 0.25$ for the whole MIREX data set, therefore the differences are not significant.

5.4 Conclusions

An efficient novel approach for onset detection has been described in this chapter. In the preprocessing stage, the spectrogram is computed and apportioned through a one semitone filter bank. The onset detection function is the normalized sum of temporal derivatives for each band, and those peaks in the detection function over a constant threshold are identified as onsets.

A simple variation has been proposed, considering adjacent frames in order to improve the accuracy for non-percussive pitched onsets. In most situations, $\tilde{o}[t]$ yields lower results than without considering additional frames, therefore it is only suitable for a few specific sounds.

The method has been evaluated and compared with other works in the MIREX (2009) audio onset detection contest. Although the system is mainly designed for tuned pitched sounds, the results are competitive for most timbral categories, except for speech or inharmonic pitched sounds.

As the abrupt harmonic variations produced at the beginning of the notes are emphasized and those produced in the sustain stage are minimized, the system performs reasonably well against smooth vibratos lower than one semitone. Therefore, $o[t]$ is suitable for percussive harmonic onsets, but it is also robust to frequency variations in the sustained sounds.

When a portamento occurs, the system usually detects a new onset when the f_0 increases or decreases more than one semitone, resulting in some false positives. However, this is not a drawback if the method is used for multiple pitch estimation.

As future work, a first categorization of the sounds could be done, like in (Tan et al., 2009). If the song belongs to the non-percussive category, $\tilde{o}[t]$ could be used instead, dynamically adjusting the value of C according to the degree of percussiveness.

An adaptive filter bank could also be included for mistuned harmonic sounds, determining the tuning like in (Zhu and Kankanhalli, 2006) and shifting the center frequencies of the bands according to the tuning deviation.

Other aspect that might be improved is the temporal resolution, by changing the window length, the hop size, or implementing a similar scheme than in (Collins, 2005a) to refine the onset position after the detection.

6

Multiple pitch estimation using supervised learning methods

A simple first approach for multiple pitch estimation in monotimbral¹ mixtures is presented in this chapter. Some of the following contents have been published in (Pertusa, 2003; Pertusa and Iñesta, 2004; Pertusa and Iñesta, 2005).

The hypothesis is that a supervised method can learn a pattern from a given timbre. The system is trained using spectrograms apportioned into a one semitone filter bank and the information about the ground-truth pitches. After the learning stage, the supervised algorithm can detect the pitches as occurrences of that pattern in the spectrogram, even in the presence of polyphony interferences (harmonic overlap). Time-delay neural networks (TDNN) and k -nearest neighbours (k NN) have been evaluated and compared for pitch classification.

The main drawback of supervised learning techniques is that they depend on the training data. The amount of timbral variety and pitch combinations in the training stage can condition the success rate. As it is difficult to get real musical data aligned with the ground-truth pitches, synthesized songs have been used in a simplified scenario, obtaining promising results for tuned synthetic sounds with fixed spectral profiles and constant amplitude envelopes. The multiple f_0 estimation of real music without any a-priori information is addressed in Chapter 7.

As previously discussed in Sec. 4.2.4, different supervised techniques have been used for multiple pitch estimation, most of them for piano sounds. It should be mentioned that the proposed methods described in the present chapter were published in 2004, and all the supervised learning approaches cited in 4.2.4 are posterior, except from the Marolt (2001) work which was the basis for (Marolt, 2004a,b).

¹Sounds produced by a single timbre.

Besides the above-mentioned methods for multiple pitch estimation, many supervised learning techniques have been used for single pitch estimation of speech and musical monophonic signals. An earlier work which is similar to the presented approach was proposed by [Taylor and Greenhough \(1993\)](#) for single pitch classification. The inputs of this method were the spectra of signals sampled at 8 kHz mapped on a distribution of one semitone bins, like in ([Sano and Jenkins, 1989](#)). Then, an adaptive resonance theory neural network called ARTMAP ([Carpenter et al., 1991](#)) was used for pitch classification, obtaining high success rates.

Time delay neural networks have also been successfully used in speech recognition problems ([Waibel, 1989](#), [Boulevard and Morgan, 1992](#)), and for multiple pitch estimation by [Marolt \(2004a\)](#). In the latter work, TDNNs were compared with multilayer perceptrons, radial basis function networks² and Elman partially recurrent networks³. Experimentally, TDNNs achieved the best results among the evaluated methods.

6.1 Preprocessing

Supervised learning techniques require a set of input features aligned with the desired outputs. In the proposed approach, a frame by frame analysis is performed, building input-output pairs at each frame. The input data are spectral features, whereas the outputs consist on the ground-truth pitches. The details of the input and output data and their construction are described in this section.

6.1.1 Construction of the input-output pairs

In order to increase the computational efficiency and to reduce the amount of spurious information in the input vectors, it is important to reduce the dimensionality of the feature set. If all the spectral bins were used, there would be 2048 or 4096 input features. Some of them are unnecessary, thus they can complicate the learning process and increase the computational complexity.

In the proposed method, like in the onset detection approach previously described, a one semitone filter bank is used to compress in some way the spectral information. As most of the spectral energy in harmonic sounds usually coincide with the center of the one semitone filters, this kind of filter bank can be used to represent the energy of the prominent harmonics, keeping the main structure of a harmonic pattern.

²A radial basis function network is an artificial neural network that uses radial basis functions as activation functions.

³The Elman network is a variation on the multilayer perceptron, with the addition of a set of context units in the input layer connected with the hidden layer units.

Input data

The training data set consists of musical audio files at $f_s = 22,050$ Hz synthesized from MIDI sequences. The STFT of the each musical piece is computed, providing the magnitude spectrogram using a 93 ms Hanning window with a 46.4 ms hop size. With these parameters, the time resolution for the spectral analysis is $\Delta t = 46.4$ ms, and the highest possible frequency is $f_s/2 = 11025$ Hz, which is high enough to cover the range of useful pitches. Like in the onset detection method, zero padding has been used to build the lower filters.

The same way as described in Chapter 5 for onset detection, the spectral values at each frame are apportioned into $B = 94$ bands using a one semitone filter bank ranging from 50 Hz ($G\sharp_0$) to $f_s/2$, almost eight octaves, yielding a vector of filtered values $\mathbf{b}[t]$ at each frame.

These values are converted into decibels and set as attenuations from the maximum amplitude, which is 96 dB⁴ with respect to quantization noise. In order to remove noise and low intensity components at each frame, a threshold ξ is applied for each band in such a way that, if $b_i[t] < \xi$, then $b_i[t] = \xi$. This threshold was empirically established at $\xi = -45$ dB. This way, the input data is within the range $\mathbf{b}[t] \in [\xi, 0]^B$.

Information about adjacent frames is also considered to feed the classifiers. For each frame at time t , the input is a set of spectral features $\{\mathbf{b}[t + j]\}$ for $j \in [-m, +n]$, being m and n the number of spectral frames considered before and after the frame t , respectively.

Output data

For each MIDI file, a binary digital piano-roll (BDP) is obtained to get the active pitches (desired output) at each frame. A BDP is a matrix where each row corresponds to a frame and each column corresponds to a MIDI pitch (see Fig. 6.1). Therefore, at each frame t , $n + m + 1$ input vectors $\mathbf{b}[t + j]$ for $j \in [-m, +n]$ and a vector of pitches $\nu[t] \in \{0, 1\}^B$ are shown to the supervised method during the training stage.

6.2 Supervised methods

Given these input-output pairs, two different supervised learning approaches (TDNN and k NN) have been evaluated and compared for pitch classification.

⁴Given the audio bit depth of the generated audio files (16 bits), the maximum amplitude in the magnitude spectrogram is $20 \log(2^{16}) = 96$ dB.

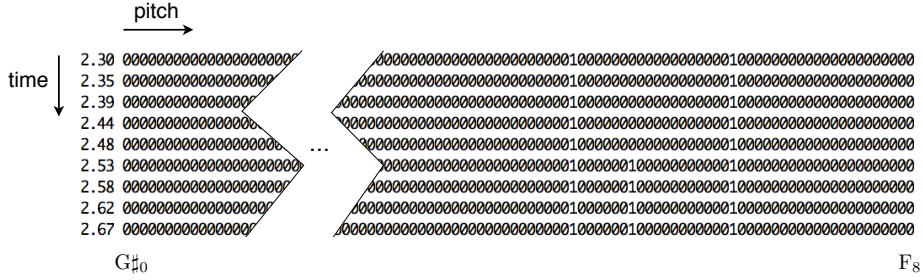


Figure 6.1: Binary digital piano-roll coding in each row the active pitches at each time frame when the spectrogram is computed.

6.2.1 Time-delay neural networks

A time-delay neural network trained with the standard backpropagation algorithm from Rumelhart et al. (1986) has been used. The scheme of the network architecture can be seen in Fig. 6.2.

The input data are normalized into the interval $[-1, +1]$, being the -1 value assigned to the maximum attenuation (ξ dB), and $+1$ the value assigned to the attenuation of 0 dB, through this simple equation:

$$\bar{b}_i[t] = \frac{1}{\xi/2}(\xi + b_i[t]) - 1 \tag{6.1}$$

This way, the input data $b_i[t] \in [\xi, 0]$ are mapped into $\bar{b}_i[t] \in [-1, +1]$ for the network input. Each of these values, which corresponds to one spectral component, is provided to a neuron at the input layer. The adjacent frames provide the short-context information. For each frame considered, B new input units are added to the network, being the total number of input neurons $B(n + m + 1)$.

The network output layer is composed of $B = 94$ neurons, one for each possible pitch. The output is coded in such a way that an activation value of $y_k[t] = 1$ for a particular unit k means that the k -th pitch is active at that frame, whereas $y_k[t] = 0$ means that the pitch is not active.

The TDNN has been implemented with bias⁵ and without momentum⁶. The selected transfer function $f(x)$ is a standard sigmoid (see Fig. 6.3):

$$f(x) = \frac{2}{1 + e^{-x}} - 1 \tag{6.2}$$

⁵A bias neuron lies in one layer, is connected to all the neurons in the next layer but none in the previous layer, and it always emits 1.

⁶Momentum, based on the notion from physics that moving objects tend to keep moving unless acted upon by outside forces, allows the network to learn more quickly when there exist plateaus in the error surface (Duda et al., 2000).

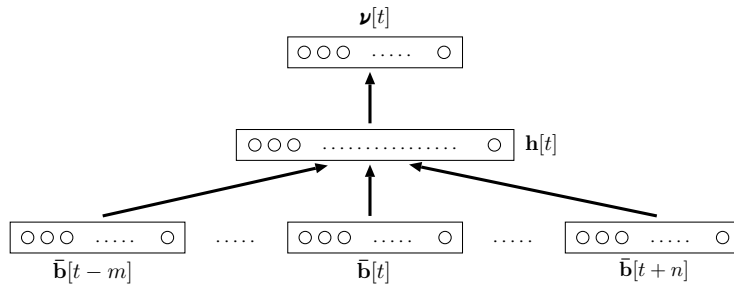


Figure 6.2: TDNN architecture and data supplied during training. The arrows represent full connection between layers.

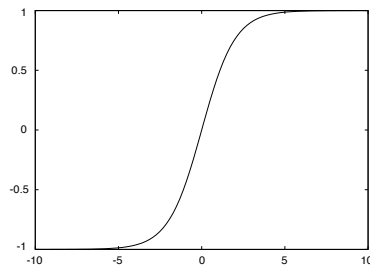


Figure 6.3: Sigmoid transfer function.

After performing the transfer function, the output values for the neurons are within the range $y_k[t] \in [-1, +1]$. A pitch is detected when $y_k[t] > \alpha$. Therefore, the activation threshold α controls the sensitivity of the network (the lower is α , the more likely a pitch is activated).

6.2.2 Nearest neighbors

In the k NN method, the vectors $\boldsymbol{\nu}[t]$ are the prototype labels. As previously discussed in Sec. 2.4.2, in the recognition stage the standard k NN algorithm can not generalize and find new prototypes not seen in the training stage. A simple extension of the k NN method has been proposed to mitigate this effect.

In the recognition stage the k nearest neighbors are identified at the target frame t , and an activation function $A_p[t]$ is obtained for each pitch p :

$$A_p[t] = \sum_{i=0}^k \nu_p^{(i)}[t] \quad (6.3)$$

being $\nu_p^{(i)}[t] \in \{0, 1\}$ the corresponding pitch p (not present/present) in the prototype $\nu^{(i)}[t]$. Then, a low level threshold ζ is established as a fraction of k , and only the pitches which accomplish $A_p[t] \geq \zeta$ in the neighboring are considered to be active at the frame t . This way, the method can infer new prototypes that are not present in the training stage.

Instead of using the number of pitch occurrences as an activation criterion, additional experiments have been done using weighted distances, summing the multiplicative inverse of the Euclidian distance $d_i[t]$ for each neighbor i to increase the importance of the pitches that are close to the test sample:

$$A'_p[t] = \sum_{i=0}^k \nu_p^{(i)}[t] \left(\frac{1}{d_i[t] + 1} \right) \quad (6.4)$$

A third activation function has been proposed, taking into account the normalized distances:

$$A''_p[t] = \sum_{i=0}^k \nu_p^{(i)}[t] \left(\frac{1}{k-1} \right) \left(1 - \frac{d_i[t]}{\sum_{\forall i} d_i[t]} \right) \quad (6.5)$$

In all these cases, if the activation function obtains a value greater than ζ , then the pitch p is added to the prototype yielded at the target frame t .

6.3 Evaluation

A data set of MIDI sequences were utilized for the evaluation of the proposed methods, obtaining input/output pairs from the MIDI files and the synthesized audio. Then, 4-folded cross-validation experiments were performed, making four subexperiments dividing the data set into four parts (3/4 for training and 1/4 for test). The presented results were obtained by averaging the subexperiments carried out on each data subset. The accuracy of the method is evaluated at frame-by-frame and note levels.

The frame (or event) level accuracy is the standard measure for multiple pitch estimation described in Eq. 3.6. A relaxed novel metric has been proposed to evaluate the system at note level. Notes are defined as series of consecutive event detections along time. A false positive note is detected when an isolated series of consecutive false positive events is found. A false negative note is defined as a sequence of isolated false negative events, and any other sequence of consecutive event detections is considered as a successfully detected note. Eq. 3.6 is also used for note level accuracy, considering false positive, false negative and correctly detected notes.

6.3.1 Generation of the training and test sets

The evaluation was done using musical pieces generated with synthetic instruments with near-constant temporal envelopes. The limitations for acoustical acquisition from real instruments played by musicians and the need of an exact timing of the ground-truth pitches have conditioned the decision for constructing these sounds using virtual synthesis models.

Polyphonic tracks of MIDI files (around 25 minutes of music) were improvised by the author and synthesized using different waveshapes, attempting to have a variety of styles and pitch combinations in the training set. In total, 2,377 different chords were present in the data set with an average polyphony of 3 simultaneous sounds. The selected timbres are described next.

Sinusoidal waveshape

This is the simplest periodic wave. Almost all the spectral energy is concentrated in the f_0 component.

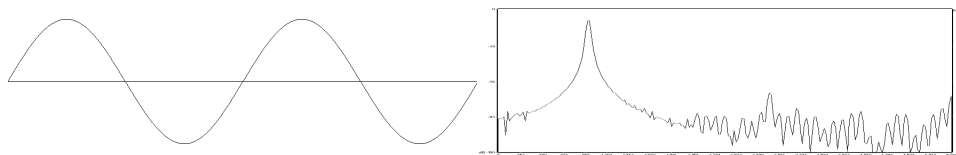


Figure 6.4: Sinusoidal waveform and spectrum in dB using a Hanning window.

Sawtooth waveshape

This sound contains all the harmonics with amplitudes proportional to $1/h$, being h the number of harmonic. Only the first $H = 10$ harmonics were used to generate this sound.

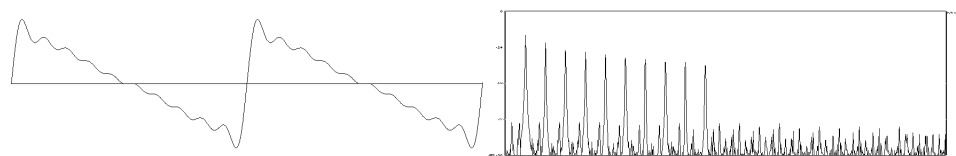


Figure 6.5: Sawtooth waveform and spectrum in dB using a Hanning window.

Clarinet waveshape

The clarinet sound is generated using a physical model of a clarinet with the *wgclar* Csound opcode, which produces good imitating synthesis.

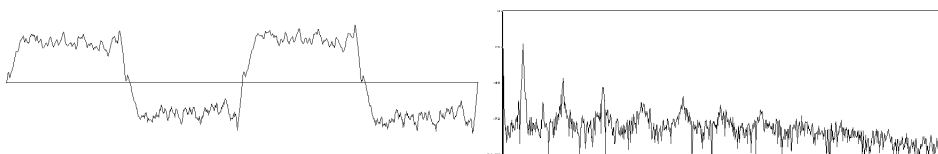


Figure 6.6: Clarinet waveform and spectrum in dB using a Hanning window.

Hammond organ waveshape

An electrophone timbre corresponding to a Hammond organ⁷ was selected. This instrument produces sound through a mechanism based on electromagnetic induction. In this work, the mechanism has also been simulated using CSound.

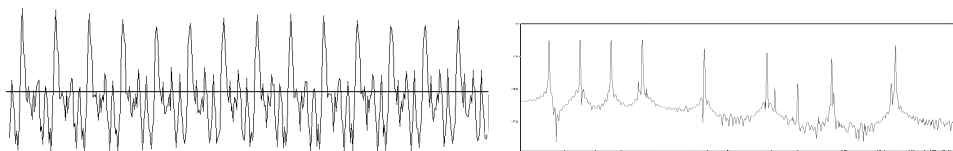


Figure 6.7: Hammond waveform and spectrum in dB using a Hanning window.

6.4 Results using time-delay neural networks

6.4.1 Neural network parametrization

Some parameters are free in any neural network. The ones of special interest in the proposed method are the number of input frames ($n + m + 1$), and the activation threshold α . The computational complexity depends on the number of input frames and, if this value is high, there can be some spectral frames merging different pitch combinations, which can difficult the training and recognition processes. The activation threshold α controls the sensitivity of the network.

These are the most relevant parameters, whereas others concerning the training, like weight initialization, number of hidden neurons, etc. have shown to be less important. Different experiments have been carried out varying these parameters and the results did not vary importantly.

The detailed parametrization results are extensively described in ([Pertusa, 2003](#)). After some initial tests, a number of hidden neurons of 100 proved

⁷An electronic organ created by L. Hammond in 1934.

	sine	sawtooth	clarinet	Hammond
events	0.94 ± 0.02	0.92 ± 0.02	0.92 ± 0.02	0.91 ± 0.02
notes	0.95 ± 0.02	0.92 ± 0.02	0.92 ± 0.02	0.92 ± 0.02

Table 6.1: Frame-by-frame and note detection accuracy using TDNN.

to be a good choice, with an initial learning rate⁸ of 0.001. The algorithm converges when the success rate does not improve during a maximum number of iterations⁹.

According to the size of the input context, the best results were obtained with $m = 1$ and $n = 0$, i.e., considering only one previous frame. Anyway, the accuracy was not much higher than that obtained with one window at each side, or even 2+1 or 1+2. The detection was consistently worse with 2+2 and larger contexts. However, the success rate was clearly worse when no context was considered, about 20% lower than with any of the other non-zero contexts tested (see (Pertusa, 2003) for details). Values of $\alpha \in [-0.8, -0.7]$ showed to be adequate.

6.4.2 Recognition results

4-folded cross-validation experiments were performed to assess the capability of the TDNN for this task with the parameters that showed consistency for the different waveshapes.

Using this data set, the training algorithm converges relatively fast (in tens of epochs), and each epoch takes about 5 seconds in a 2 GHz iMac.

The Table 6.1 shows the accuracy and dispersion¹⁰ of note and event detection for the timbres described in Sec. 6.3.1. As expected, the sinusoidal waveshape provided the best results (around 0.94 for events and 0.95 for notes). Most of the event detection errors were false negatives at the note boundaries, and the majority of the errors in note detection corresponded to false negative notes shorter than 100 ms.

Using the sawtooth timbre, the success rates are lower due to the higher harmonic content. Around 0.92 for events and also for notes were obtained. Again, most of the false negatives occurred in very short notes.

For the clarinet and the Hammond organ the results were comparable to those obtained for the pure synthetic waveshapes, giving values around 0.92 for notes and ranging from 0.90 to 0.92 for events. The results obtained with

⁸An adequate learning rate must be selected to ensure that the weights converge to a response that is neither too specific nor too general.

⁹This parameter has been set to 15 iterations.

¹⁰Dispersion is calculated as the difference between the maximum and the minimum accuracy from the 4 subexperiments divided by 4.

6. MULTIPLE PITCH ESTIMATION USING SUPERVISED LEARNING

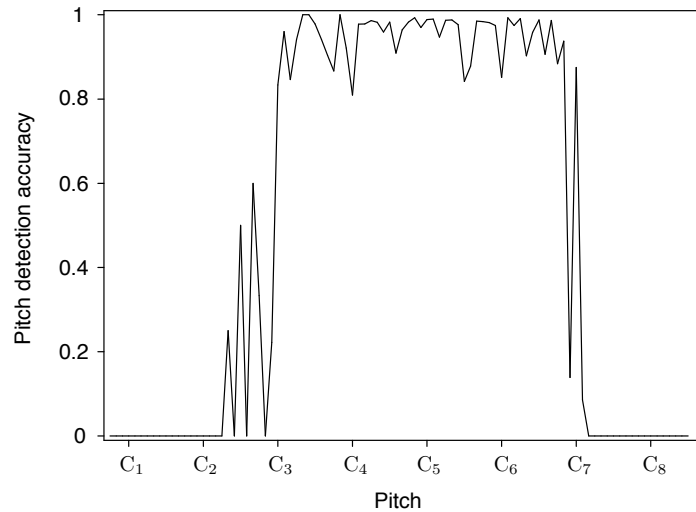


Figure 6.8: TDNN recognition accuracy as a function of pitch.

the clarinet and the Hammond suggest that the methodology can be applied to other instruments characterized by a nearly stable amplitude envelope.

The errors have been analyzed considering note length, pitch, and number of training samples. Errors produced by notes shorter than 100 ms represent the 31% of the total amount of errors. With a time resolution $\Delta t = 46$ ms, these notes extend along one or two frames. Since most of the false negatives occur at the beginning and end of the notes, these very short notes, which are not usual in real music, are sometimes missed.

As shown in Fig. 6.8, most of pitch errors correspond to very high (higher than C7) and very low (lower than C3) pitches which are very unfrequent in real music, whereas the method has a very high success rate in the central range of pitches. This effect is partially related with the amount of musical pitches in the training set, which is composed of musical data. The most frequent musical pitches are those at valid central frequencies. There exists a clear correlation of recognition success for a given pitch to the amount of events in the training set for that pitch. In Fig. 6.9, each dot represents a single pitch. Abcises represent the amount of data for that pitch in the training set, whereas ordinates represent the recognition accuracy. An exponential curve has been adjusted to the data showing the clear non linear correlation between the amount of training data and the performance.

Another reason to explain these errors is that lowest pitches are harder to detect due to the higher frequency precision required, and highest pitches have less harmonics below Nyquist. Moreover, the harmonics of highest pitches can

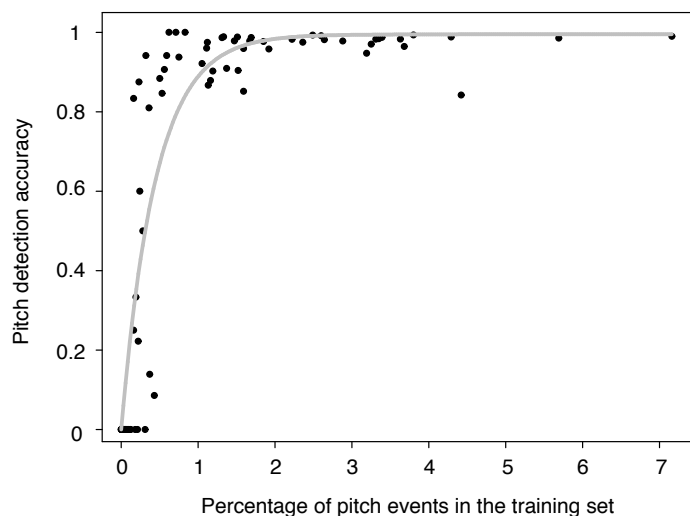


Figure 6.9: TDNN correlation between recognition rates for each pitch and the amount of events in the training set for that pitch.

also produce aliasing when they are synthesized. Anyway, most of the wrong estimates correspond to very unusual notes that were artificially introduced in the data set to spread out the pitch range, and which are not common in real music.

A graphical example of the detection is shown in Fig. 6.10. This musical excerpt was neither in the training set nor in the recognition set. It was synthesized using the clarinet timbre¹¹, and with a fast tempo (120 bpm). In this example, the event detection accuracy was 0.94, and most of the errors were produced in the note onsets or offsets. Only 3 very short false positive notes were detected.

6.4.3 Changing waveshapes for detection

The results for the evaluated waveshapes were similar, showing that the performance does not critically depends on the selected timbre, at least for instruments with a fixed spectral profile. To assess how specific the network weights are for the different timbres considered, musical pieces generated with a given timbre were presented to a network trained with a different instrument. The event and note detection results are displayed in tables 6.2 and 6.3, respectively.

¹¹Clarinet training weights were also used for recognition.

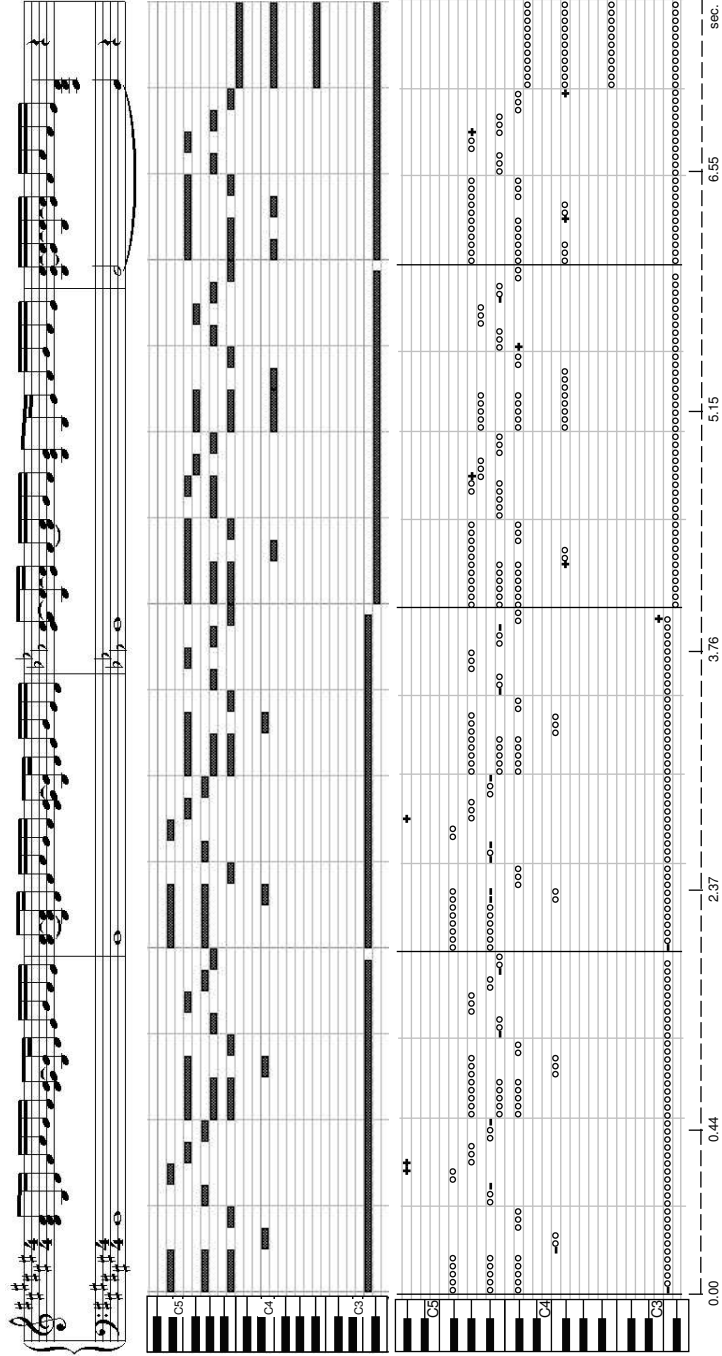


Figure 6.10: Temporal evolution of the note detection for a given melody using the clarinet timbre. Top: the original score; center: the melody as displayed in a sequencer piano-roll; down: the piano-roll obtained from the network output compared with the original piano-roll. Notation: ‘o’: successfully detected events, ‘+’: false positives, and ‘-’: false negatives.

Events	sine	sawtooth	clarinet	Hammond
sine	0.94 ± 0.02	0.57 ± 0.03	0.70 ± 0.04	0.30 ± 0.02
sawtooth	0.48 ± 0.02	0.92 ± 0.02	0.69 ± 0.02	0.26 ± 0.03
clarinet	0.46 ± 0.03	0.67 ± 0.03	0.92 ± 0.02	0.34 ± 0.02
Hammond	0.083 ± 0.014	0.169 ± 0.009	0.144 ± 0.007	0.91 ± 0.02

Table 6.2: Frame level cross-detection results using TDNN. Rows correspond to training timbres and columns to test timbres.

Notes	sine	sawtooth	clarinet	Hammond
sine	0.95 ± 0.02	0.46 ± 0.02	0.61 ± 0.06	0.27 ± 0.02
sawtooth	0.51 ± 0.03	0.92 ± 0.02	0.65 ± 0.01	0.29 ± 0.02
clarinet	0.57 ± 0.04	0.56 ± 0.02	0.92 ± 0.02	0.337 ± 0.008
Hammond	0.089 ± 0.014	0.164 ± 0.009	0.140 ± 0.002	0.92 ± 0.02

Table 6.3: Note level cross-detection results using TDNN. Rows correspond to training timbres and columns to test timbres.

The accuracy ranges from 0.089 to 0.65 for pitch recognition of sounds which are different from those used to train the TDNN, showing the network specialization. The cross-detection value could be an indication of the similarity between two timbres, but this assumption needs of further in-deep study.

6.5 Results using k nearest neighbors

Initial experiments were done using only 1NN, i.e., classifying the sample with the closest prototype label. Then, different values for k and ζ have been evaluated using the three k NN activation functions previously described.

To directly compare the k NN results with the TDNN, the spectral information of the previous frame ($m = 1$, $n = 0$) was also added to the input.

The evolution of the event detection accuracy using the sinusoidal waveshape with the different activation functions is shown in Figs. 6.11, 6.12, and 6.13. These figures are representative of the system behavior with the different timbres.

In the first activation function A_p , values of ζ have been set as fractions of k , to make them relative to the number of neighbors. The same occurs with A'_p , but in this case different fractions have been considered, as ζ depends on the distances, instead of the pitch counts. For the normalized function A''_p , the threshold values vary from 0 to 1.

The experiments using these thresholds and changing the number of neighbors have been done for all the timbres, obtaining the event and note accuracy for each of them. In the case of A_p , values of $\zeta = k/2$ and $\zeta = 2k/3$

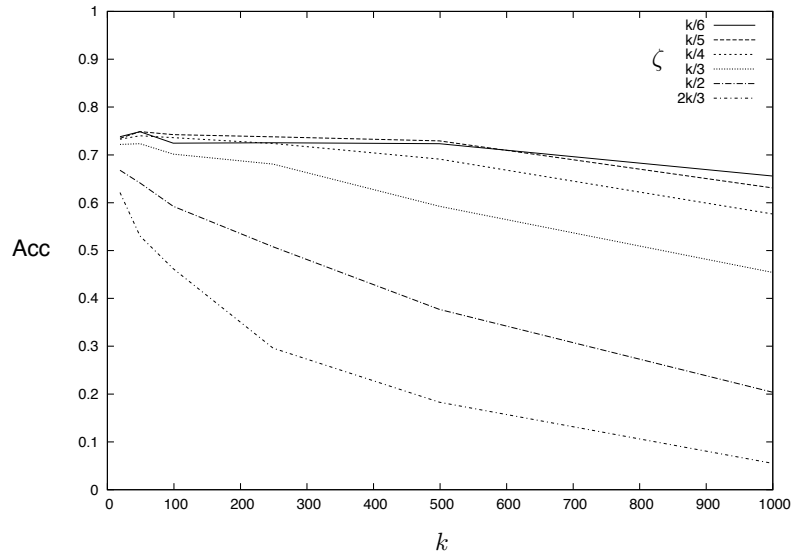


Figure 6.11: Event detection accuracy using A_p for the sinusoidal timbre with respect to k and ζ .

have provided the best results for event and note detection (see Tabs. 6.4 and 6.5). When k becomes large, the accuracy decreases, and good values for k are relatively small (from 20 to 50). The behavior is similar for all the tested timbres. In most cases, A_p obtains a significantly higher accuracy than when using only one nearest neighbor.

No significant differences were found comparing the best results for A'_p and A_p . However, when using A'_p , the number of neighbors does not affect much the results (see 6.12). The highest accuracy was obtained with $\zeta \in \{k/200, k/300\}$.

The best results for most timbres were obtained using A''_p with $k = 20$. Tabs. 6.4 and 6.5 show the success rate for events and notes using $k = 20$, which is the best k value among those tested for most timbres (except from the sinusoidal waveshape, where $k = 50$ yielded a slightly higher accuracy). It can be seen that A''_p obtains the highest accuracy for most timbres. Anyway, the best results are significantly worse than those obtained using the TDNN.

6.6 Conclusions

In this chapter, different supervised learning approaches for multiple pitch estimation have been presented. The input/output pairs have been generated by sequencing a set of MIDI files and synthesizing them using CSound. The magnitude STFT apportioned through one semitone filter-bank is used as input

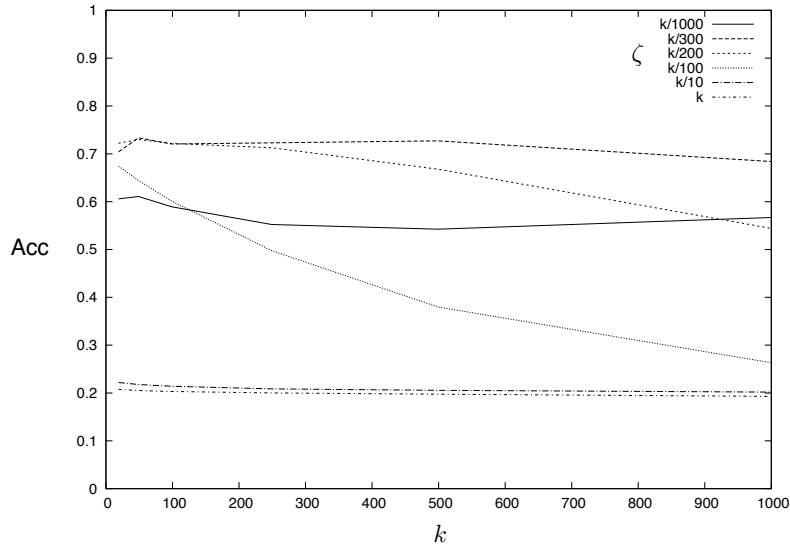


Figure 6.12: Event detection accuracy using A'_p for the sinusoidal timbre with respect to k and ζ .

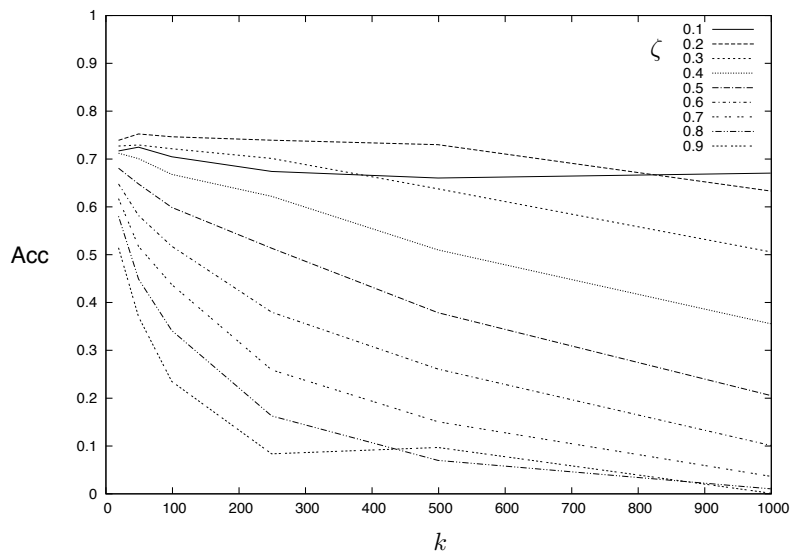


Figure 6.13: Event detection accuracy using A''_p for the sinusoidal timbre with respect to k and ζ .

6. MULTIPLE PITCH ESTIMATION USING SUPERVISED LEARNING

Events	k	ζ	sinusoidal	sawtooth	clarinet	Hammond
1-NN			0.67 ± 0.02	0.62 ± 0.02	0.50 ± 0.02	0.604 ± 0.011
A_p	20	$k/6$	0.74 ± 0.02	0.644 ± 0.011	0.49 ± 0.02	0.637 ± 0.014
	20	$k/5$	0.74 ± 0.02	0.653 ± 0.010	0.48 ± 0.02	0.638 ± 0.011
	20	$k/4$	0.73 ± 0.02	0.654 ± 0.007	0.50 ± 0.02	0.643 ± 0.010
	20	$k/3$	0.722 ± 0.014	0.653 ± 0.007	0.49 ± 0.02	0.641 ± 0.010
	20	$k/2$	0.668 ± 0.012	0.633 ± 0.011	0.49 ± 0.02	0.612 ± 0.014
	20	$2k/3$	0.62 ± 0.02	0.61 ± 0.02	0.48 ± 0.02	0.58 ± 0.02
A'_p	20	$k/1000$	0.61 ± 0.03	0.54 ± 0.03	0.457 ± 0.014	0.55 ± 0.02
	20	$k/300$	0.70 ± 0.02	0.616 ± 0.014	0.49 ± 0.02	0.62 ± 0.02
	20	$k/200$	0.72 ± 0.02	0.637 ± 0.006	0.49 ± 0.02	0.632 ± 0.010
	20	$k/100$	0.67 ± 0.02	0.633 ± 0.009	0.47 ± 0.03	0.60 ± 0.02
	20	$k/10$	0.22 ± 0.03	0.19 ± 0.03	0.17 ± 0.03	0.16 ± 0.03
	20	k	0.21 ± 0.03	0.09 ± 0.03	0.04 ± 0.02	0.08 ± 0.02
A''_p	20	0.1	0.72 ± 0.03	0.62 ± 0.02	0.48 ± 0.02	0.62 ± 0.02
	20	0.2	0.74 ± 0.02	0.651 ± 0.011	0.49 ± 0.02	0.644 ± 0.013
	20	0.3	0.727 ± 0.013	0.655 ± 0.007	0.50 ± 0.02	0.643 ± 0.010
	20	0.4	0.712 ± 0.013	0.654 ± 0.007	0.49 ± 0.02	0.638 ± 0.007
	20	0.5	0.681 ± 0.012	0.637 ± 0.011	0.49 ± 0.02	0.623 ± 0.011
	20	0.6	0.648 ± 0.012	0.623 ± 0.013	0.49 ± 0.02	0.60 ± 0.02
	20	0.7	0.62 ± 0.03	0.60 ± 0.02	0.48 ± 0.02	0.58 ± 0.02
	20	0.8	0.58 ± 0.02	0.58 ± 0.02	0.47 ± 0.03	0.54 ± 0.03
	20	0.9	0.51 ± 0.03	0.53 ± 0.02	0.44 ± 0.03	0.49 ± 0.03

Table 6.4: Event accuracy using $k = 20$ for each activation function.

Notes	k	ζ	sinusoidal	sawtooth	clarinet	Hammond
1-NN			0.65 ± 0.02	0.51 ± 0.02	0.46 ± 0.04	0.530 ± 0.013
A_p	20	$k/6$	0.71 ± 0.02	0.55 ± 0.03	0.44 ± 0.04	0.55 ± 0.02
	20	$k/5$	0.722 ± 0.009	0.57 ± 0.03	0.45 ± 0.04	0.57 ± 0.02
	20	$k/4$	0.728 ± 0.008	0.59 ± 0.03	0.47 ± 0.04	0.59 ± 0.02
	20	$k/3$	0.728 ± 0.009	0.60 ± 0.03	0.48 ± 0.04	0.60 ± 0.02
	20	$k/2$	0.705 ± 0.008	0.614 ± 0.013	0.49 ± 0.04	0.617 ± 0.010
	20	$2k/3$	0.675 ± 0.008	0.612 ± 0.010	0.49 ± 0.05	0.613 ± 0.007
A'_p	20	$k/1000$	0.38 ± 0.03	0.29 ± 0.02	0.41 ± 0.03	0.33 ± 0.03
	20	$k/300$	0.61 ± 0.03	0.46 ± 0.03	0.42 ± 0.03	0.50 ± 0.03
	20	$k/200$	0.688 ± 0.008	0.52 ± 0.03	0.45 ± 0.03	0.57 ± 0.03
	20	$k/100$	0.70 ± 0.02	0.596 ± 0.012	0.45 ± 0.03	0.605 ± 0.009
	20	$k/10$	0.35 ± 0.09	0.30 ± 0.08	0.25 ± 0.08	0.26 ± 0.05
	20	k	0.33 ± 0.09	0.18 ± 0.06	0.09 ± 0.04	0.15 ± 0.05
A''_p	20	0.1	0.62 ± 0.03	0.46 ± 0.03	0.39 ± 0.05	0.47 ± 0.03
	20	0.2	0.720 ± 0.010	0.56 ± 0.03	0.45 ± 0.04	0.56 ± 0.02
	20	0.3	0.731 ± 0.008	0.59 ± 0.03	0.48 ± 0.04	0.59 ± 0.02
	20	0.4	0.733 ± 0.008	0.61 ± 0.02	0.48 ± 0.04	0.62 ± 0.02
	20	0.5	0.717 ± 0.007	0.611 ± 0.013	0.49 ± 0.04	0.622 ± 0.013
	20	0.6	0.697 ± 0.007	0.618 ± 0.010	0.49 ± 0.05	0.620 ± 0.008
	20	0.7	0.671 ± 0.011	0.614 ± 0.012	0.48 ± 0.05	0.617 ± 0.008
	20	0.8	0.65 ± 0.02	0.600 ± 0.012	0.47 ± 0.05	0.605 ± 0.011
	20	0.9	0.60 ± 0.03	0.57 ± 0.02	0.45 ± 0.06	0.57 ± 0.03

Table 6.5: Note accuracy using $k = 20$ for each activation function.

data, whereas the outputs are the ground-truth MIDI pitches. Two different supervised learning methods (TDNN and k NN) have been used and compared for this task using simple stationary sounds and taking into account adjacent spectral frames.

The TDNN performed far better than the k NN, probably due to the huge space of possible pitch combinations. The results suggest that the neural network can learn a pattern for a given timbre, and it can find it in complex mixtures, even in the presence of beating or harmonic overlap. The success rate was similar in average for the different timbres tested, independently of the complexity of the pattern, which is one of the points in favour of this method.

The performance using the nearest neighbors is clearly worse than the TDNN approach. Different alternatives were proposed to generalize in some way the prototypes matched using the k NN technique to obtain new classes (pitch combinations) not seen in the training stage. However, these modifications did not improve significantly the accuracy. An interesting conclusion from this comparison is that k NN techniques are not a good choice for classification when there exists many different prototype labels, as in this particular task.

Respect to the TDNN method, errors are concentrated in very low/high frequencies, probably due to the sparse presence of these pitches in the training set. This fact suggests that increasing the size and variety of the training set, the accuracy could be improved. In the temporal dimension, most of the errors are produced in the note boundaries, which are not very relevant from a perceptual point of view. This is probably caused by the window length, which can cover transitions between different pitch combinations. When the test waveshape was different from that used to train the net, the recognition rate decreased significantly, showing the high specialization of the network.

The main conclusions are that a TDNN approach can estimate accurately the pitches in simple waveforms, and the compact input using a one semitone filter-bank is representative of the spectral information for harmonic pitch estimation.

Future work include to test the feasibility for this approach for real mixtures of sounds with varying temporal envelopes, but this requires of a large labeled data set for training, and it is difficult to get musical audio pieces perfectly synchronized with the ground-truth pitches. However, this is a promising method that should be deeply investigated with real data.

It also seems reasonable to provide the algorithm with a first timbre recognition stage, at least at instrument family level. This way, different weight sets could be loaded in the net according to the decision taken by the timbre recognition algorithm before starting the pitch estimation.

7

Multiple fundamental frequency estimation using signal processing methods

Supervised learning methods require synchronized audio and symbolic data to be trained. These methods rely on the training set, and for this reason most of them need a-priori information about the timbre to be transcribed. Probably, it could be possible to generalize and correctly find the pitches in real audio files when they are trained using a large data set, but they still depend on the training data.

In this chapter, three multiple f_0 estimation methods are described. These heuristics approaches purely rely on signal processing methods, avoiding the use of a training set. The first of them is a simple and efficient iterative cancellation approach, mainly intended for percussive strings sounds¹. The method was integrated into a more complex genre classification system published in (Lidy et al., 2007) to get a basic estimation with a very low computational cost.

Besides the iterative cancellation approach, two novel joint estimation methods have been proposed. As discussed in Sec. 4.3, joint estimation methods can model the source interactions better than iterative cancellation approaches. However, they tend to have higher computational costs due to the evaluation of many possible source combinations. Contrary to most joint estimation methods, the proposed techniques have been designed to be very efficient. They were evaluated and compared to other works in MIREX (2007) and MIREX (2008) multiple f_0 estimation and tracking contests, yielding competitive results with very efficient runtimes. The first joint estimation approach is detailed in (Pertusa and Iñesta, 2007, 2008a), and the second in (Pertusa and Iñesta, 2008b).

¹Here, the term percussive string instrument sound is used to refer to the sounds of plucked and struck string instruments such as piano, guitar and pizzicato violin.

The methods described in this chapter are implemented in C++, and they can be compiled and executed from the command line in Linux and Mac OSX. Two standard C++ libraries have been used, for loading the audio files (libsndfile²), and computing the Fourier transforms (FFTW3 from Frigo and Johnson (2005)). The rest of code, including the generation of MIDI files, has been implemented by the author.

7.1 Iterative cancellation method

A simple and efficient iterative cancellation method is described in this section. The general scheme of the system is shown in Fig. 7.1. In the preprocessing stage, the STFT is computed, and the sinusoidal components are extracted using a sinusoidal likeness measure (SLM). The onsets are detected using the method described in Chapter 5. Then, only the frames that are after each detected onset are analyzed to yield the active fundamental frequencies in the interval between two consecutive onsets.

At each analyzed frame, a set of f_0 candidates are selected from the sinusoidal spectral peaks. The candidates are evaluated in ascending frequency order. For each candidate, the partials are first searched, and a fixed harmonic pattern is used to subtract the candidate spectral components from the mixture using iterative cancellation. Finally, a postprocessing stage is done to refine the estimate, removing pitches with a very low absolute or relative intensity.

The proposed iterative cancellation method is mainly intended for piano transcription, as the selected harmonic pattern is based on an approximation to the spectra of most percussive string sounds.

7.1.1 Preprocessing

In the preprocessing stage, the magnitude spectrogram is obtained performing the STFT with a 93 ms Hanning-windowed frame and a 46 ms hop size. This window size may seem long for typical signal processing algorithms, but for chord identification pitch margin is wide (Klapuri, 2003a), and this is also the frame length used in many previous methods for multiple f_0 estimation, like (Klapuri, 2006b; Rynänen and Klapuri, 2005). Using these spectral parameters, the temporal resolution achieved is 46 ms. Zero padding has been used, multiplying the original size of the window by a factor z to complete it with zeroes before computing the STFT. With this technique, the frequency of the lower pitches can be more precisely estimated.

Then, at each frame, a sinusoidal likeness measure (SLM) is calculated to identify the spectral peaks that correspond to sinusoids, allowing to discard the

²<http://www.mega-nerd.com/libsndfile/>

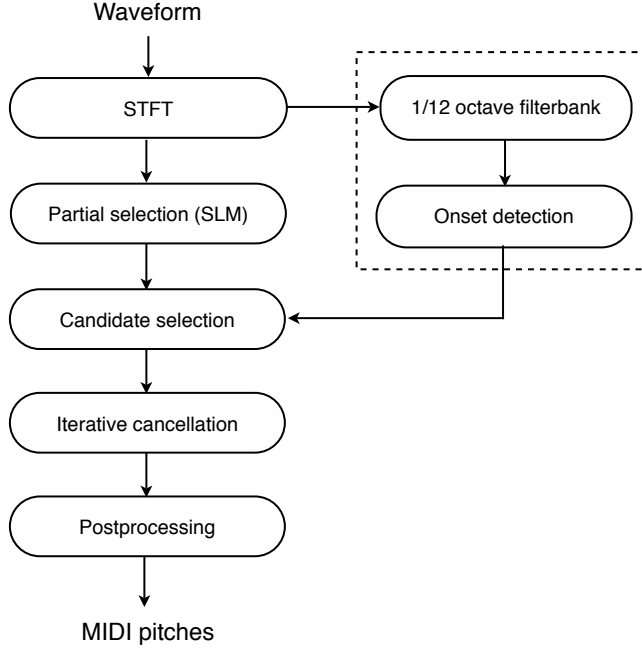


Figure 7.1: Scheme of the proposed iterative cancellation method.

spurious non-sinusoidal components. SLM techniques have been successfully applied to the analysis and synthesis of speech (Griffin and Lim, 1985) and musical sounds (Doval, 1994, Rodet, 1997, Virtanen, 2000).

The SLM from Rodet (1997) provides a measure v_Ω of the likeness between a pure sinusoid and a partial candidate with frequency Ω . As pointed out by Virtanen (2000), basically the idea is to calculate the cross-correlation between the short-time spectrum of the signal and the spectrum resulting from an ideal sinusoid, scaling the result by the overall spectral shape.

Being $H(\omega)$ the DFT of the analysis window³, and $X(\omega_k)$ the amplitude corresponding to the frequency ω_k , the cross-correlation function Γ of the complex signals H and X within a bandwidth⁴ W can be calculated as:

$$\Gamma(\omega) = \sum_{k, |\omega - \omega_k| < W} H(\omega - \omega_k) X(\omega_k) \quad (7.1)$$

Each maximum $|\Gamma(\Omega)|$ indicates a sinusoidal partial candidate at frequency Ω . Defining norms for $H(\omega_k)$ and $X(\omega_k)$ for a frequency Ω :

³In the proposed method, a Hanning function.

⁴SLM assumes that there are not two sinusoids closer than W Hz.

7. MULTIPLE F_0 ESTIMATION USING SIGNAL PROCESSING METHODS

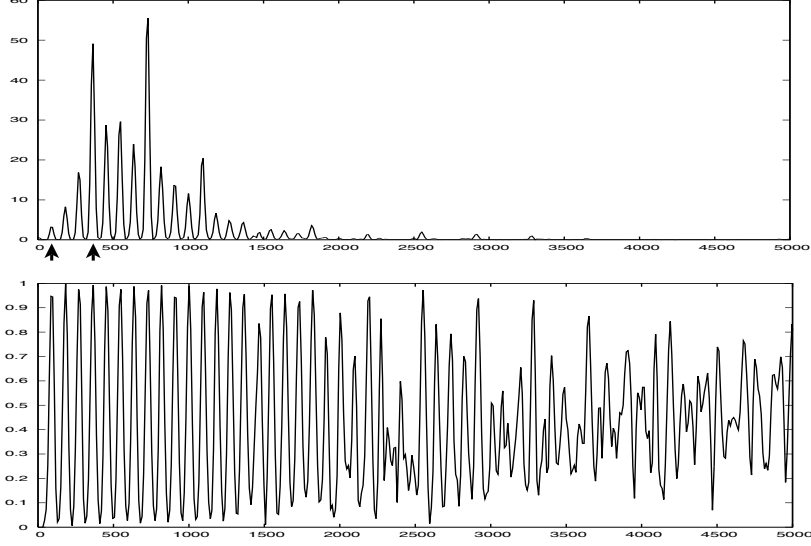


Figure 7.2: Example magnitude spectrum (top) and SLM (bottom) for two sounds in an octave relation (92.5 and 370 Hz) using $W = 50$ Hz. The fundamental frequencies are indicated with arrows.

$$|H|_{\Omega}^2 = \sum_{k, |\Omega - \omega_k| < W} |H(\Omega - \omega_k)|^2 \quad (7.2)$$

$$|X|_{\Omega}^2 = \sum_{k, |\Omega - \omega_k| < W} |X(\omega_k)|^2 \quad (7.3)$$

the SLM at a frequency Ω can be computed as:

$$v_{\Omega} = \frac{|\Gamma(\Omega)|}{|H|_{\Omega}|X|_{\Omega}} \quad (7.4)$$

An efficient implementation of v_{Ω} has been chosen using the method proposed by [Virtanen \(2000\)](#). The cross-correlation of the frequency domain is performed through a multiplication of the time-domain signals, and $\Gamma(\omega)$ is calculated using the DFT for $x[n]$ windowed twice with the Hanning function. The calculation of $|X|_{\Omega}^2$ can be implemented with a IIR filter which has only two non-zero coefficients: one delay takes a cumulative sum of the signal and the other subtracts the values at the end of the window.

After this process, a sinusoidal likeness function (see Fig. 7.2) is obtained at each frame. The harmonic peak selection is done as follows: if there is a peak in the SLM function which value is $v_{\Omega} > \tau$, being τ a constant threshold,

then the original spectral component at the same frequency Ω , with its original amplitude, is added to the harmonics list. The spectral components that do not satisfy the previous condition are discarded. Therefore, the mid-level representation of the proposed method consists on a sparse vector containing only certain values of the original spectrum (those ideally corresponding to partials). This sparse representation reduces the computational cost with respect to the analysis of all spectral peaks.

7.1.2 Onset detection

The onsets are detected from the STFT using the method described in Chapter 5, identifying each frame at t_i as onset or not-onset. For efficiency, only a single frame between two consecutive onsets is analyzed to yield the fundamental frequencies within that inter-onset interval.

To avoid the analysis of transients at the onset times, the frame chosen to detect the active notes is $t_o + 1$, being t_o the frame where an onset was detected. Therefore, only those frames that are 46 ms after a detected onset are analyzed to estimate the fundamental frequencies in the interval between two adjacent onsets. The scheme for estimating the pitches only between two consecutive onsets has also been used in the recent method from [Emiya et al. \(2008b\)](#).

7.1.3 Candidate selection

For each frame $t_o + 1$, a selection of f_0 candidates is done from the sparse array of sinusoidal peaks. Therefore, as many other iterative cancellation approaches, the method assumes that the partial corresponding to the fundamental frequency is present. This assumption is made to improve the efficiency, as it reduces significantly the number of candidates respect to the analysis of the whole spectral range.

There are two restrictions for a peak to be a candidate; only candidates within a given pitch margin $[f_{min}, f_{max}]$ are considered, and the difference between the candidate frequency and the frequency of the closest pitch in the equal temperament must be lower than f_d Hz⁵. This is a constant value introduced to remove some false candidates at high frequencies.

7.1.4 Iterative cancellation

The candidates are sorted in ascending frequency order, and they are evaluated using the iterative cancellation method described in Alg. 1.

The partials of each candidate are searched taking inharmonicity into account. Similarly to ([Bello et al., 2002](#)), ([Every and Szymanski, 2006](#)), and

⁵This technique is not suitable for not tuned instruments.

Algorithm 1: Iterative cancellation method

Input: Sinusoidal peaks and f_0 candidates at a given frame**Output:** Estimated fundamental frequencies

```

residual  $\leftarrow$  sinusoidal peaks
for each  $f_0$  candidate do
  intensity  $\leftarrow$   $f_0$  amplitude
  for  $h = 2$  to  $H$  do
    find partial  $h$  in residual
    if partial not found then
      | missingPartials  $\leftarrow$  missingPartials + 1
    else
      | expected  $\leftarrow$   $f_0$  amplitude  $\cdot$   $\mathbf{p}[h - 1]$ 
      | obtained  $\leftarrow$  partial amplitude
      | if expected > obtained then
      | | Remove partial from the residual
      | else
      | | residual[partial]  $\leftarrow$  residual[partial]  $-$  expected
      | end
    end
    intensity  $\leftarrow$  intensity + expected
    if missingPartials > allowed missing partials then
      | Discard candidate
    end
  end
  if Candidate is discarded then
    | Residual is not updated
  else
    | Add candidate to the  $f_0$  list
  end
end

```

(Emiya et al., 2008b), the partial frequency deviation model proposed by Fletcher and Rossing (1988) for piano sounds, and previously described in Eq. 2.17, has been used with $\beta = 0.0004$ (Fletcher and Rossing, 1988). A harmonic h is identified as the closest spectral peak to hf_0 within the margin $[hf_0 \pm f_h]$.

When a sinusoidal peak is identified as a candidate partial, its expected magnitude⁶ is calculated according to a fixed harmonic pattern \mathbf{p} and then subtracted from the residual. The chosen harmonic pattern represents the relative amplitudes of the first $H = 8$ harmonics with respect to the amplitude of the f_0 with the following weights:

⁶i.e., the expected contribution to the mixture.

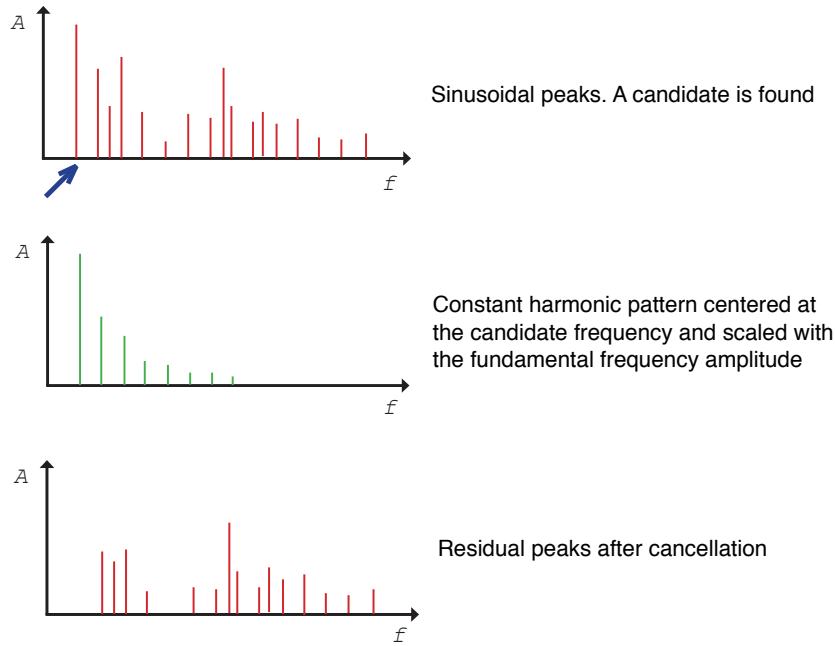


Figure 7.3: Candidate cancellation example.

$$\mathbf{p} = \{1, 0.5, 0.4, 0.3, 0.2, 0.1, 0.05, 0.01\} \quad (7.5)$$

Therefore, the harmonic pattern is a vector where each component represents a harmonic amplitude relative to the f_0 amplitude, and the first component is always 1. The selected pattern is adequate for percussive string instruments, as it approximates the spectral envelope of a piano sound and has a similar sawtooth shape to the $1/h$ model proposed by Klapuri (2006b).

When a partial is found, its expected amplitude is set as the f_0 amplitude multiplied by $\mathbf{p}[h - 1]$. If this value is greater than the actual spectral peak amplitude, the sinusoidal peak is completely removed. Otherwise, the expected amplitude is linearly subtracted from the corresponding spectral peak (see Fig. 7.3), thus assuming the additivity of linear spectrum.

Candidates without a minimum number of found harmonics experimentally set as $H/2$ are discarded. The intensity l_n of a note is finally calculated as the sum of the expected harmonic amplitudes.

7.1.5 Postprocessing

Those candidates with a low absolute or relative intensity are removed. First, the pitch candidates with a intensity $l_n < \gamma$ are discarded. The maximum note intensity $L = \max_{\forall n} \{l_n\}$ at the target frame is calculated to remove the candidates with $l_n < \eta L$, as the sources in the mixture should not have very important energy differences⁷.

Finally, the frequencies of the selected candidates are converted to MIDI pitches with Eq. 2.21. Using this inter-onset based scheme, there are certain ambiguous situations that are not produced in a frame by frame analysis. If a pitch is detected in the current and previous inter-onset interval, then there are two possibilities: there exists a single note spanning both onsets, or there is a new note with the same pitch.

To make a simple differentiation between new notes and detections of pitches that were already sounding in the previous frames, the estimation is done at frames $t_o + 1$, and $t_o - 1$. If a detected pitch at frame $t_o + 1$ is not detected at $t_o - 1$, then a new note is yielded. Otherwise, the note is considered to be a continuation of the previous estimate.

7.2 Joint estimation method I

A novel joint evaluation method for multiple f_0 estimation is presented in this section. In contrast to most previous joint approaches, this method has a very low computational cost keeping a high accuracy.

The overall scheme of the method can be seen in Alg. 2. The system performs a frame by frame analysis, yielding a set of pitch estimates at each instant. A set of pitch candidates are first identified from the spectral peaks at each frame⁸. All the possible candidate combinations are generated, and a joint algorithm is used to find the best combination taking into account the source interactions.

To evaluate a combination, a hypothetical partial sequence (HPS⁹) is built for each candidate. A candidate score (salience) is calculated taking into account the sum of its HPS harmonic amplitudes and the smoothness measure of the spectral envelope. The salience of a combination is computed as the sum of the squared candidate saliences, and the combination with highest salience is selected at the target frame.

The method assumes that the spectral envelopes of the analyzed sounds tend to vary smoothly as a function of frequency. The spectral smoothness principle

⁷When this occurs, the notes with lower energy are hardly perceived, being sometimes masked by the other harmonic components in the mixture.

⁸In the proposed joint estimation methods, SLM is not used.

⁹HPS is the term proposed by Yeh et al. (2005) to refer a vector containing hypothetical partial amplitudes (see pag. 65).

Algorithm 2: Joint estimation method I

Input: Spectral peaks at a given frame**Output:** Estimated fundamental frequencies

Select candidates from spectral peaks
 Generate all possible candidate combinations
for each *combination* \mathcal{C} **do**
 residual \leftarrow spectral peaks
 for each *candidate* $c \in \mathcal{C}$ **do**
 estimate HPS from residual
 residual \leftarrow residual $-$ HPS amplitudes
 evaluate HPS salience
 end
 combination salience \leftarrow sum of squared HPS saliences
end
return combination with max salience

has also been used in different ways in the literature (Klapuri, 2003a, Yeh et al., 2005, Cañadas-Quesada et al., 2008, and Zhou et al., 2009). The proposed novel smoothness measure is based on the convolution of the hypothetical harmonic pattern with a gaussian window.

Given a combination, the HPS of each candidate is calculated considering the harmonic interactions with the partials of all the candidates in the combination. The overlapped partials are first identified, and their amplitudes are estimated by linear interpolation using the non-overlapped harmonic amplitudes.

In contrast with the previous iterative cancellation method, which assumes a constant harmonic pattern, the proposed joint approach can estimate hypothetical harmonic patterns from the spectral data, evaluating them according to the properties of harmonic sounds. This approach is suitable for most real harmonic sounds, in contrast with the iterative method, which assumes a constant pattern based in percussive string instruments.

7.2.1 Preprocessing

In the preprocessing stage, the STFT is computed using a 93 ms Hanning windowed frame, with a 9.28 ms hop size. The frame overlap ratio may seem high from a practical point of view, but it was required to compare the method with other works in the MIREX (2007) evaluation contest (see Sec. 7.4.3). Like in the iterative cancellation method, zero padding has been used to get a more precise estimation of the lower frequencies.

SLM has not been used in the joint estimation approaches. Experimentally, including SLM in the iterative cancellation algorithm did not improve the results (see Sec. 7.4.1 for details), so it was removed in the joint estimation methods

that were implemented subsequently. Instead, a simple peak picking algorithm was used to extract the hypothetical partials from the magnitude spectrum. At each frame, the spectral peaks with an amplitude higher than a given constant threshold μ are selected, discarding the rest of spectral information and obtaining this way a sparse representation which only contains certain spectral bins for the following analysis.

7.2.2 Candidate selection

The evaluation of many combinations of candidates increases significantly the computational cost of the algorithm. To improve the efficiency, the candidates are first ordered decreasingly by the sum of their harmonic amplitudes and, at most, only the first F candidates of this ordered list are considered. The adequate selection of candidates plays an important role in joint estimation methods, and a good compromise between the accuracy and computational cost (which depends on the number of candidates) must be chosen.

Like in the iterative cancellation approach, the f_0 candidates are selected from the spectral peaks¹⁰ that fulfill a series of requirements. First, a candidate must be within the range $[f_{min}, f_{max}]$, which corresponds to the pitches of interest. If a spectral peak amplitude is lower than a threshold ε , then the peak is discarded as a candidate.

To search for the partials, a constant margin $f_h \pm f_r$ around each harmonic frequency is considered, allowing slight harmonic deviations. The closest peak to the center of this margin is set as a found partial. If there are no spectral peaks within this margin, a missing harmonic is considered.

Like in (Yeh, 2008), the harmonic spectral location and spectral interval principles (Klapuri, 2004) have been considered, taking inharmonicity into account. The ideal frequency f_h of the first partial is initialized to $f_h = 2f_0$. The next partials are searched at $f_{h+1} = (f_x + f_0) \pm f_r$, where $f_x = f_i$ if the previous harmonic h was found at the frequency f_i , or $f_x = f_h$ if the previous partial was not found.

In some methods, the candidate selection is done according to harmonicity criterion (Yeh, 2008), partial beating (Yeh, 2008), or the product of harmonic amplitudes in the power spectrum (Emiya et al., 2008b). The proposed method uses the sum of harmonic amplitudes as a score function for candidate selection, avoiding that finding a harmonic with an amplitude lower than 1 will decrease the score, as it occurs when the harmonic magnitudes are multiplied.

¹⁰As the candidates are spectral peaks, timbres with missing fundamental are not considered, like in the iterative cancellation method.

7.2.3 Generation of combinations of candidates

All the possible candidate combinations are calculated and evaluated, and the combination with highest salience is selected at the target frame. The combinations consist of different number of pitches. In contrast with other works, like (Yeh et al., 2005), there is not need for a-priori estimation of the number of concurrent sounds before detecting the fundamental frequencies, and the polyphony is implicitly calculated in the f_0 estimation stage, selecting the combination with highest score independently from the number of candidates.

At each frame, a set of combinations $\{\mathcal{C}_1, \mathcal{C}_2, \dots, \mathcal{C}_N\}$ is obtained. For efficiency, like in the recent approach from Emiya et al. (2008b), only the combinations with a maximum polyphony P are generated from the F candidates. The amount of combinations N without repetition can be calculated as:

$$N = \sum_{n=1}^P \binom{F}{n} = \sum_{n=1}^P \frac{F!}{n!(F-n)!} \quad (7.6)$$

This means that when the maximum polyphony is $P = 6$ and there are $F = 10$ selected candidates, $N = 847$ combinations are generated. Therefore, N combinations are evaluated at each frame, and the adequate selection of F and P is critical for the computational efficiency of the algorithm.

7.2.4 HPS estimation

The candidates of a combination \mathcal{C} are ordered by ascending frequency. Then, a HPS vector \mathbf{p}_c , consisting on the hypothetical harmonic amplitudes of the first H harmonics, is estimated for each candidate $c \in \mathcal{C}$:

$$\mathbf{p}_c = \{p_{c,1}, p_{c,2}, \dots, p_{c,h}, \dots, p_{c,H}\} \quad (7.7)$$

where $p_{c,h}$ is the amplitude of the h harmonic of the candidate c . The partials are searched as previously described for the candidate selection stage. If a particular harmonic is not found, then the corresponding value $p_{c,h}$ is set to zero.

Once the partials of a candidate are identified, the HPS values are estimated considering hypothetical source interactions. To do it, the harmonics of all the candidates in the combination are first identified, and they are labeled with the candidate they belong to (see Fig. 7.4). After the labeling process, there are harmonics that only belong to one candidate (non-overlapped harmonics), and harmonics belonging to more than one candidate (overlapped harmonics).

Assuming that the interactions between non-coincident partials (beating) do not alter significantly the original spectral amplitudes, the non-overlapped

7. MULTIPLE F_0 ESTIMATION USING SIGNAL PROCESSING METHODS

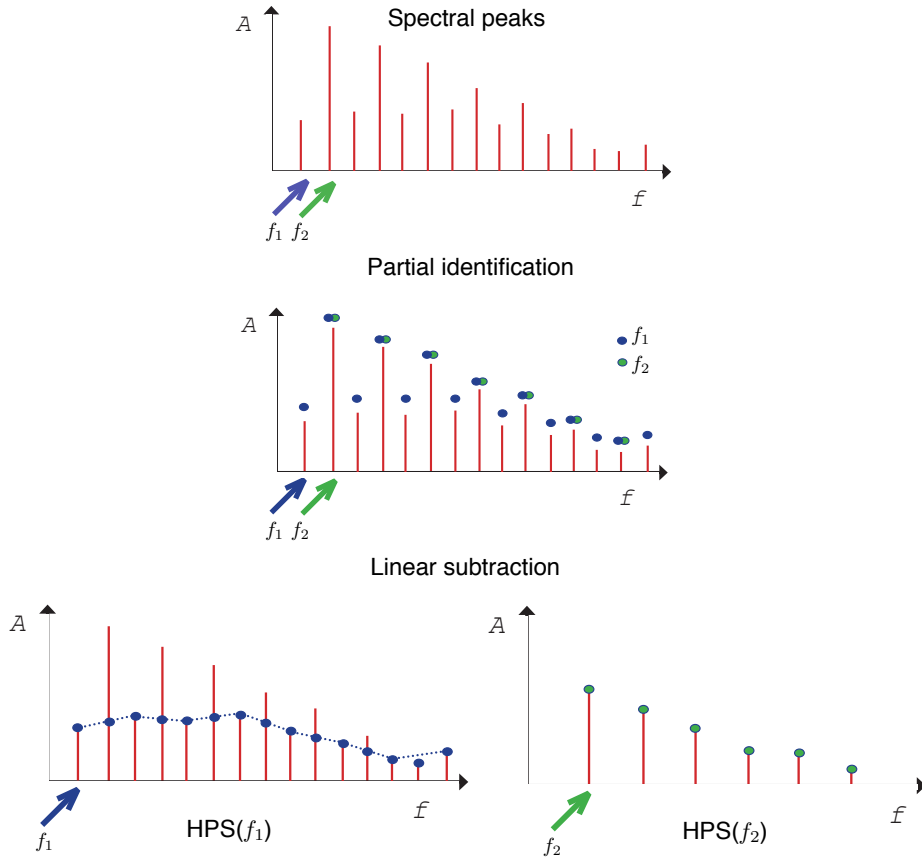


Figure 7.4: Interpolation example in a combination of two candidates separated by an octave. The HPS are estimated with the interpolated values.

amplitudes are directly assigned to the HPS. However, the contribution of each source to an overlapped partial amplitude must be estimated. This can be done using the amplitudes of non-overlapped neighbor partials (Klapuri, 2003a, Yeh et al., 2005, Every and Szymanski, 2006), assuming smooth spectral envelopes, or considering that the amplitude envelopes of different partials are correlated in time (Woodruff et al., 2008).

In the proposed method, similarly to (Maher, 1990) and (Yeh et al., 2005), the amplitudes of overlapped partials are estimated by linear interpolation of the neighboring non-overlapped partials (see Fig. 7.4).

If there are more than two consecutive overlapped partials, then the interpolation is done the same way with the non-overlapped values. For instance, if harmonics 2 and 3 are overlapped, then the amplitudes of harmonics 1 and 4 are used to estimate them by linear interpolation.

After the interpolation, the contribution of each partial to the mixture must be estimated and subtracted before processing the next candidates. This calculation is done as follows:

- If the interpolated (expected) value is greater than the corresponding overlapped harmonic amplitude, then $p_{c,h}$ is set as the original harmonic amplitude, and the spectral peak is completely removed from the residual, setting it to zero for the candidates that share that partial.
- If the interpolated value is smaller than the corresponding overlapped harmonic amplitude, then $p_{c,h}$ is set as the interpolated amplitude, and this value is linearly subtracted for the candidates that share the harmonic (see Fig. 7.4).

Therefore, the additivity of linear spectrum ($\cos(\phi_\Delta) = 0$) is assumed to estimate the amplitude of colliding partials. The residual harmonic amplitudes after this process are iteratively analyzed for the rest of the candidates in the combination in ascending frequency order.

7.2.5 Salience of a combination

Once the HPS of all the candidates have been estimated for a given combination, their saliences are calculated. The salience of a combination is the squared sum of the saliences of its candidates. A candidate salience is obtained taking into account the intensity and the smoothness of its HPS.

The intensity $l(c)$ of a candidate c is a measure of the strength of a source, and it is computed as the sum of the HPS amplitudes:

$$l(c) = \sum_{h=1}^H p_{c,h} \quad (7.8)$$

Like in other works, the method also assumes that a smooth spectral pattern is more probable than an irregular one. To compute the smoothness σ of a candidate, the HPS is first normalized dividing the amplitudes by the maximum harmonic value in the HPS, obtaining $\bar{\mathbf{p}}$. Then, $\bar{\mathbf{p}}$ is low-pass filtered using a truncated normalized Gaussian window $\mathcal{N}_{0,1}$, which is convolved with the HPS to obtain the smoothed version $\tilde{\mathbf{p}}$:

$$\tilde{\mathbf{p}}_c = \mathcal{N}_{0,1} * \bar{\mathbf{p}}_c \quad (7.9)$$

Only three components were chosen for the Gaussian window $\mathcal{N} = \{0.21, 0.58, 0.21\}$, due to the reduced size of the HPS¹¹.

¹¹Usually, only the first harmonics contain most of the energy of a harmonic source, therefore typical values for H are within the margin $H \in [5, 20]$.

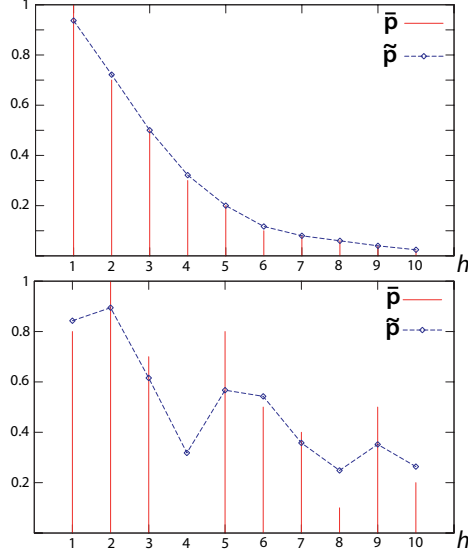


Figure 7.5: Spectral smoothness measure example. The normalized HPS vector \mathbf{p} and the smooth version $\tilde{\mathbf{p}}$ of two candidates (c_1 , c_2) are shown. Sharpness values are $s(c_1) = 0.13$, and $s(c_2) = 1.23$.

Then, as shown in Fig. 7.5, a sharpness measure $s(c)$ is computed by summing the absolute differences between the smoothed values and the normalized HPS amplitudes:

$$s(c) = \sum_{h=1}^H (|\tilde{\mathbf{p}}_{c,h} - \bar{\mathbf{p}}_{c,h}|) \quad (7.10)$$

The sharpness $s(c)$ is normalized into $\bar{s}(c)$:

$$\bar{s}(c) = \frac{s(c)}{1 - \mathcal{N}_{0,1}(\bar{x})} \quad (7.11)$$

And finally, the smoothness $\sigma(c) \in [0, 1]$ of a HPS is calculated as:

$$\sigma(c) = 1 - \frac{\bar{s}(c)}{H_c} \quad (7.12)$$

where H_c is the index of the last harmonic found for the candidate. This parameter was introduced to prevent that high frequency candidates that have less partials than those at low frequencies will have higher smoothness. This way, the smoothness is considered to be more reliable when there are more partials to estimate it.

Once the smoothness and the intensity of each candidate have been calculated, the salience $S(\mathcal{C}_i)$ of a combination \mathcal{C}_i with C candidates is:

$$S(\mathcal{C}_i(t)) = \sum_{c=1}^C [l(c) \cdot \sigma^\kappa(c)]^2 \quad (7.13)$$

where κ is a factor to control the smoothness weight.

Note that when there are overlapped partials, their amplitudes are estimated by interpolation, and the HPS smoothness tends to increase. To partially compensate this effect in the salience function, the candidate saliences are squared in Eq. 7.13 before summing them. This way, the method assigns a higher score to a combination of one candidate with a given salience than to a combination of two candidates with half salience values. The square factor favors a sparse representation, as it is convenient to explain the mixture with the minimum number of sources. Experimentally, it was found that this square factor was very important to improve the success rate of the method (see pag. 147).

The combination with highest salience is selected at the target frame. Similarly to the iterative cancellation approach, combinations that have at least one candidate with $l(c) < \eta$ or $l(c) < \gamma L$ are discarded, being $L = \max_{\forall c} \{l(c)\}$ the maximum intensity of the candidates in the combination.

7.2.6 Postprocessing

After selecting the best combination at each individual frame, a last stage is applied to remove some local errors taking into account the temporal dimension. If a pitch was not detected in a target frame but it was found in the previous and next frames, it is considered to be active in the current frame too, avoiding some temporal discontinuities. Notes shorter than a minimum duration d are also removed.

Finally, the sequences of consecutive detected fundamental frequencies are converted into MIDI pitches. The maximum intensity of the entire song candidates is used as reference to get the MIDI velocities, linearly mapping the candidate intensities within the range $[0, \max_{\forall c} \{l(c)\}]$ into MIDI values $[0, 127]$.

7.3 Joint estimation method II

In the previous joint estimation method, each frame was independently analyzed, yielding the combination of fundamental frequencies which maximizes a salience measure. One of the main limitations of this approach is that the selected window size (93 ms) is relatively short to perceive the pitches in a

complex mixture, even for expert musicians. As discussed in Sec. 3.1, context is very important in music to disambiguate certain situations. The joint estimation method II is an extension of the previous method, but considering information about adjacent frames, similarly to the supervised learning method described in Chapter 6, producing a smoothed detection across time.

7.3.1 Temporal smoothing

In this method, instead of selecting the combination with highest salience at isolated frames, adjacent frames are also considered to get the salience of each combination of pitches, performing a temporal smoothing.

In order to merge similar information across time, the frequencies of each combination \mathcal{C} are first converted into pitches using Eq. 2.21, obtaining a pitch combination \mathcal{C}' . For instance, the combination $\mathcal{C}_i = \{261 \text{ Hz}, 416 \text{ Hz}\}$ is mapped into $\mathcal{C}'_i = \{C_3, G\sharp_3\}$. If there are more than one combination with the same pitches at a target frame, it is removed, keeping only the combination with the highest salience value.

Then, at a target frame t , a new smoothed salience function $\tilde{S}(\mathcal{C}'_i(t))$ for a combination \mathcal{C}'_i is computed using the neighbor frames:

$$\tilde{S}(\mathcal{C}'_i(t)) = \sum_{j=t-K}^{t+K} S(\mathcal{C}'_i(j)) \quad (7.14)$$

This way, the saliences of the combinations with the same pitches than \mathcal{C}'_i in the K adjacent frames are summed to obtain the salience at the target frame, as shown in Fig. 7.6. The combination with maximum salience is finally selected to get the pitches at the target frame t .

$$\mathcal{C}'(t) = \arg \max_i \{\tilde{S}(\mathcal{C}'_i(t))\} \quad (7.15)$$

This new approach increases the robustness of the system in the data set used for evaluation, and it allows to remove the minimum amplitude ε for a peak to be a candidate, added in the previous approach to avoid local false positives.

If the selected combination at the target frame does not contain any pitch (if there is not any candidate or if none of them can be identified as a pitch), then a rest is yielded without evaluating the combinations in the K adjacent frames.

This technique smoothes the detection in the temporal dimension. For a visual example, let's consider the smoothed intensity of a given candidate c' as:

$$\tilde{l}(c'(t)) = \sum_{j=t-K}^{t+K} l(c'(j)) \quad (7.16)$$

Combinations at $t - 1$, t and $t + 1$ using $K = 1$. Best combination at frame t using method I is highlighted

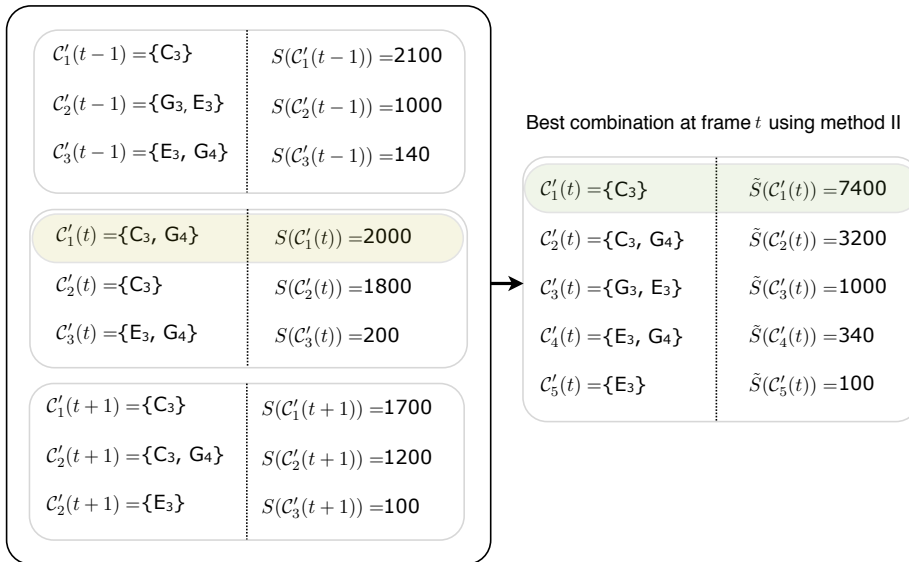


Figure 7.6: Example of combinations merged across adjacent frames using $K = 1$. The selected combination differs using methods I and II.

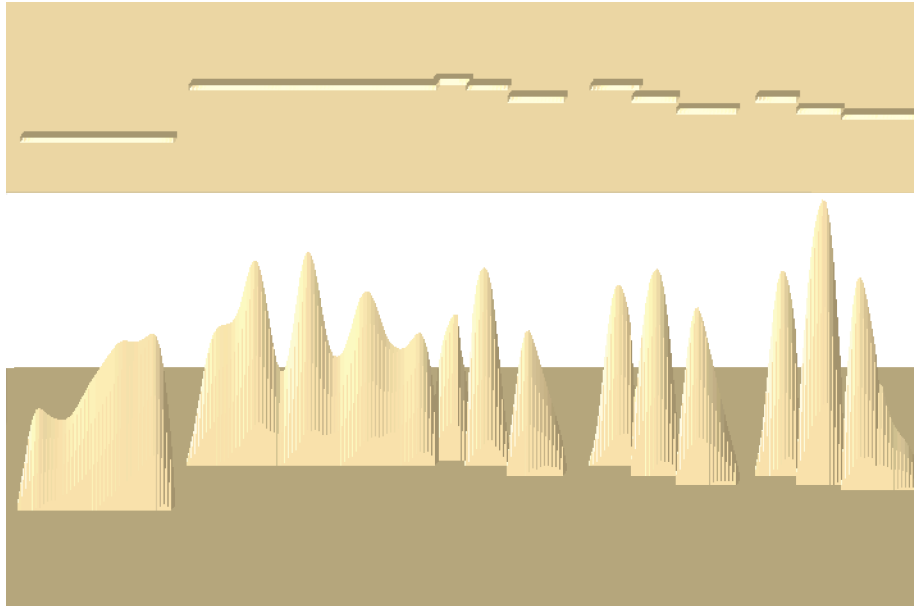


Figure 7.7: Top: Example of detected piano-roll for an oboe melody. Bottom: Three-dimensional temporal representation of $\tilde{l}(c'(t))$ for the candidates of the winner combination at each frame. In this example, all the pitches were correctly detected. High temporal smoothness usually indicates good estimates.

When the temporal evolution of the smoothed intensities $\tilde{l}(c'(t))$ of the winner combination candidates is plotted in a three-dimensional representation (see Figs. 7.7 and 7.8), it can be seen that the correct estimates usually show smooth temporal curves. An abrupt change (a sudden note onset or offset, represented by a vertical cut in the smoothed intensities 3D plot) means that the harmonic components of a given candidate were suddenly assigned to another candidate in the next frame. Therefore, vertical lines in the plot usually indicate errors mapping harmonic components with the candidates.

7.3.2 Partial search

Like in the joint estimation method I, a partial is searched at $f_{h+1} = (f_x + f_0) \pm f_r$, being $f_x = f_i$ if the previous harmonic h was found at the frequency f_i , or $f_x = f_h$ otherwise.

However, in the method II, instead of just selecting the closest peak, a triangular window has been used. This window, centered in f_h with a bandwidth $2f_r$ and a unity amplitude, has been used to weight the partial magnitudes within this range (see Fig. 7.9). The spectral peak with maximum weighted

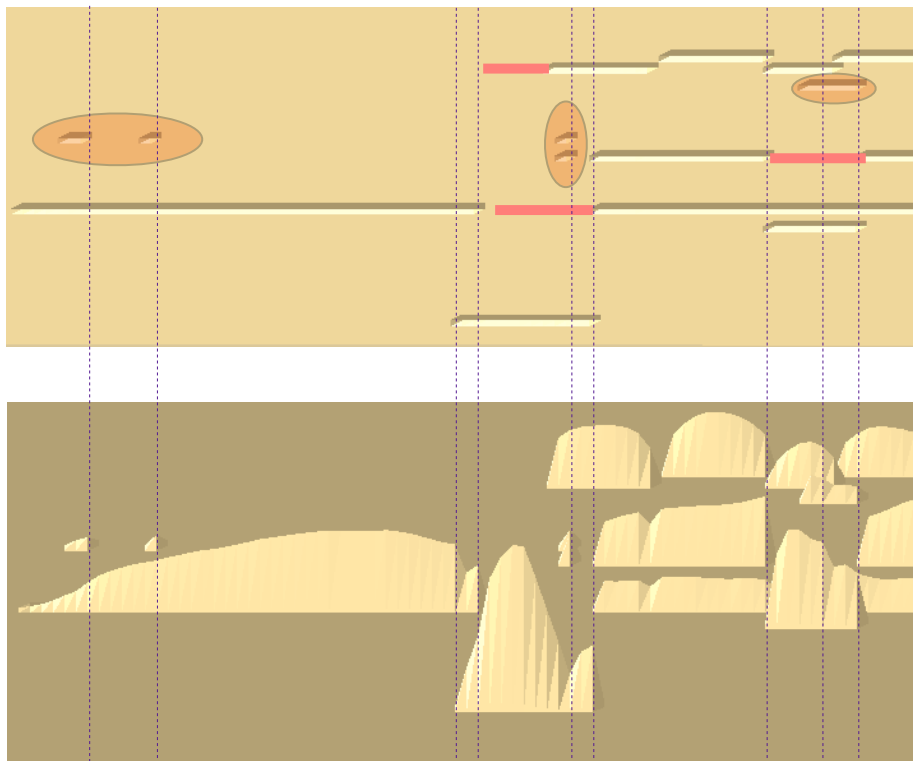


Figure 7.8: Top: Example of detected piano-roll for a mixture of sounds. False positives are marked with circles and false negatives with horizontal red bars. Bottom: Three-dimensional temporal representation of $\tilde{l}(c'(t))$ for the candidates of the winner combination at each frame. Note that most errors occur when there exist vertical transitions which occur when harmonics are suddenly reassigned to another candidate.

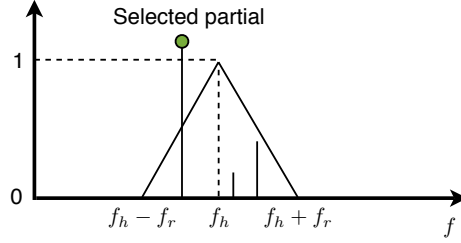


Figure 7.9: Partial selection in the joint estimation method II. The selected peak is the one with the greatest weighted value.

value is selected as a partial. The advantage of this scheme is that low amplitude peaks are penalized and, besides the harmonic spectral location, intensity is also considered to identify the most important spectral peaks with partials.

7.3.3 Fundamental frequency tracking

A simple f_0 tracking technique has been introduced to increase the temporal coherence using a weighted directed acyclic graph (wDAG). The idea is to apply a penalty when there exist abrupt changes in the estimate, favoring this way smoother temporal detections. This is done by measuring the differences in the intensities of the candidates between two consecutive frames.

Let $G = (V, E, w)$ be a wDAG, with vertex set V , edge set E , and edge function w , where $w(v_i, v_j)$ is the weight of the edge from the vertex v_i to v_j . Each vertex $v_i \in V$ represents a combination \mathcal{C}'_i . The vertices are organized in layers (see Fig. 7.10). Each layer V_t contains the M combinations with highest salience at a time frame t .

The edges connect all the vertices of a layer with all the vertices of the next layer, in a way that, if $(v_i, v_j) \in E$, then $v_i \in V_t$ and $v_j \in V_{t+1}$. The weights $w(v_i, v_j)$ between two combinations are computed as follows:

$$w(v_i, v_j) = \frac{D(v_i, v_j)}{S(v_j) + 1} \quad (7.17)$$

where $S(v_j)$ is the salience of the combination at vertex v_j and $D(v_i, v_j)$ is a similarity measure for two combinations v_i and v_j , corresponding to the sum of the absolute differences between the intensities of all the candidates in both combinations:

$$D(v_i, v_j) = \sum_{\forall c \in v_i, v_j} |\tilde{l}(v_{i,c}) - \tilde{l}(v_{j,c})| + \sum_{\forall c \in v_i - v_j} \tilde{l}(v_{i,c}) + \sum_{\forall c \in v_j - v_i} \tilde{l}(v_{j,c}) \quad (7.18)$$

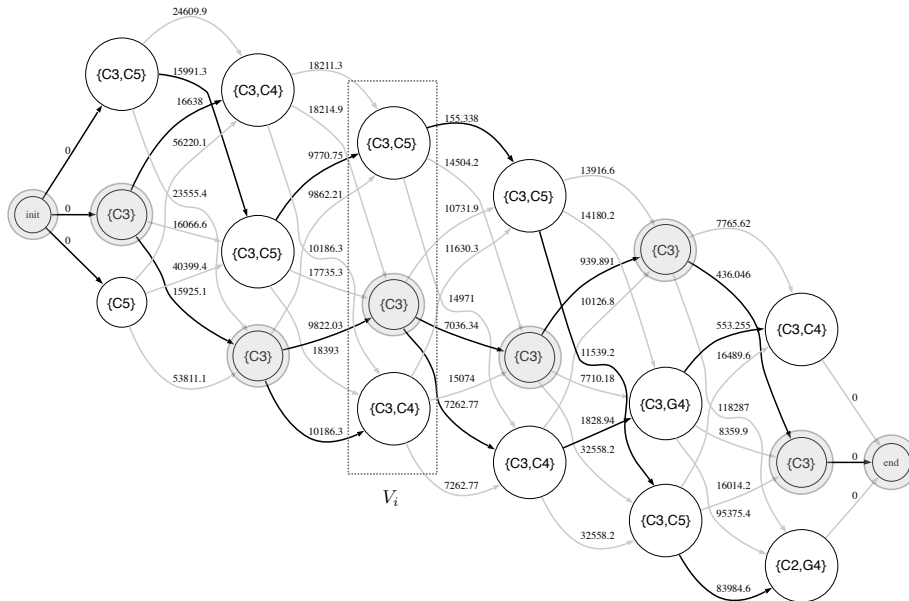


Figure 7.10: wDAG example for $M = 3$ combinations at each time frame. Each vertex represents a combination C'_i . The vertices are organized in columns which correspond to the V_i layers. Weights have been multiplied by 10^6 for visual clarity. The shadowed nodes are the pitch combinations selected at each frame. In this example, all the pitches were correctly detected.

Using this scheme, the transition weight between two combinations considers the salience of the target combination and the differences between the candidate intensities.

Finally, the shortest path¹² across the wDAG is found using the [Dijkstra \(1959\)](#) algorithm¹³. The vertices that belong to the shortest path are the winner combinations yielded at each time frame.

7.3.4 Alternative architectures

When using the joint estimation method II to identify the fundamental frequencies, there are different new alternative architectures that can be considered:

1. Frame by frame analysis. All the frames are analyzed to yield the estimates. This is the basic scheme of the joint estimation methods previously described.

¹²The path which minimizes the weights sum from the starting node to the final state.

¹³The boost C++ library, available at <http://www.boost.org>, was used for this task.

7. MULTIPLE F_0 ESTIMATION USING SIGNAL PROCESSING METHODS

2. To detect onsets and analyze only one frame between two onsets to yield the pitches in the inter-onset interval. This scheme, used in the iterative estimation method, increases the efficiency but with an accuracy cost. The method relies on the onset detection results, therefore a wrong estimate in the onset detection stage can affect the results.
3. To detect onsets and merge combinations of those frames that are between two consecutive onsets, yielding the pitches for the inter-onset interval. This technique can obtain more reliable results when the onsets are correctly estimated, as it happens with piano sounds. However, merging combinations between two frames reduce the number of detected notes, as only combinations that are present in most of the IOI frames are considered. Like in the previous scheme, the detection is very sensitive to false negative onsets.
4. To detect beats and merge combinations with a quantization grid. Once the beats are estimated¹⁴, a grid split with a given beat divisor $1/q$ can be assumed, considering that there are not triplets and that the minimum note duration is q . For instance, if $q = 4$, each inter-beat interval can be split in q sections, each one of a one sixteenth length. Then, the combinations of the frames that belong to the quantization unit can be merged to obtain the results at each minimum grid unit. Like in the onset detection scheme, the success rate of this approach depends on the success rate of beat estimation.

The implementation of the joint estimation method II allows to run the algorithm using any of these schemes. The adequate choice of the scheme depends on the signal to be analyzed. For instance, for percussive timbres, it is recommended to use the third scheme, as usually onset detection is very reliable for this kind of sounds. These architectures have been perceptually evaluated using some example real songs, but rigorous evaluation of these schemes is left as future work, since an aligned dataset of real musical pieces with symbolic data is required for this task.

In order to obtain a more readable score, the tempo changes can optionally be written into the output MIDI file. To do it, the system allows to load a list of beat times. A tempo $\mathcal{T} = 60/T_b$ is reestimated at each beat instant using the temporal difference T_b between the current and the previous beat. There is not other metrical information extracted, therefore the bar impulses are sometimes shifted due to anacrusis¹⁵ or incorrect time signature, which is always assumed to be of 4/4, since this is the most frequently used musical meter.

¹⁴Beats can be estimated with an external beat tracking algorithm like BeatRoot from Dixon (2006).

¹⁵Like it occurs in Fig. 1.1.

7.4 Evaluation

To perform a first evaluation and set up the parameters, initial experiments were done using a data set of random mixtures. Then, the three proposed approaches were evaluated and compared with other works for real music transcription in the [MIREX \(2007\)](#) and [MIREX \(2008\)](#) multiple f_0 estimation and tracking contests.

7.4.1 Parametrization

The parameters of the three proposed methods and their impact on the results are analyzed in this section. The intention in the parametrization stage is not to get the parameter values that maximize the accuracy for the test set used, as the success rate is dependent on these particular data. However, this stage can help to obtain a reasonable good parameter set of values and to evaluate the impact of each parameter in the accuracy and the computational cost. Therefore, the selected parameters are not always those that achieve the highest accuracy in the test set, but those that obtain a close-to-best accuracy keeping a low computational cost.

For the parametrization stage, a database of random pitch combinations has been used. This database was generated using mixtures of musical instrument samples with fundamental frequencies ranging between 40 and 2100 Hz. The samples are the same used in the evaluation of the [Klapuri \(2006b\)](#) method.

The data set consists on 4000 mixtures with polyphony¹⁶ 1, 2, 4, and 6. The 2842 audio samples from 32 musical instruments used to generate the mixtures are from the McGill University master samples collection¹⁷, the University of Iowa¹⁸, IRCAM studio online¹⁹, and recordings of an acoustic guitar. In order to respect the copyright restrictions, only the first 185 ms of each mixture²⁰ were used for evaluation.

It is important to note that the data set only contains isolated pitch combinations, therefore the evaluation of the parameters that have a temporal dimension (like the minimum note duration) could not be evaluated using this database. The test set is intended for evaluation of multiple f_0 estimation at single frames, therefore f_0 tracking from joint estimation method II could not be evaluated with these data.

To evaluate the parameters in the iterative cancellation method and in the joint estimation method I, only one frame which is 43 ms apart from the

¹⁶There are 1000 mixtures for each polyphony.

¹⁷<http://www.music.mcgill.ca/resources/mums/html/index.htm>

¹⁸<http://theremin.music.uiowa.edu/MIS.html>

¹⁹<http://forumnet.ircam.fr/402.html?&L=1>

²⁰Thanks to A. Klapuri for providing this reduced data set for evaluation.

7. MULTIPLE F_0 ESTIMATION USING SIGNAL PROCESSING METHODS

Stage	Parameter	Symbol	Value
Preprocessing	SLM bandwidth	W	50 Hz
	SLM threshold	τ	0.1
	Zero padding factor	z	8
Candidate selection	f_0 range	$[f_{min}, f_{max}]$	[38, 2100] Hz
	closest pitch distance	f_d	3 Hz
Postprocessing	Absolute intensity threshold	γ	5
	Relative intensity threshold	η	0.1

Table 7.1: Iterative cancellation method parameters. Shadowed parameters (W and τ) were removed after the initial experiments, as SLM did not improve the accuracy.

beginning of the mixture has been selected. For the joint estimation method II, which requires more frames for merging combinations, all the frames (5) have been used to select the best combination in the mixture.

The accuracy metric (Eq. 3.6) has been chosen as a success rate criterion for parametrization. A candidate identification error rate was also defined for adjusting the parameters that are related with the candidate selection stage. This error rate is set as the number of actual pitches that are not present in the candidate set divided by the number of actual pitches.

The overall results for the three methods using the random mixtures data set and the selected parameters are described in Sec. 7.4.2, and the results of the comparison with other multiple f_0 estimation approaches are detailed in Secs. 7.4.3 and 7.4.4.

Iterative cancellation method

The parameters chosen for the iterative cancellation method are shown in Tab. 7.1.

In the SLM analysis (see pag. 121), the threshold τ was set to a very low value, as it is preferable to have a noise peak than discarding a partial in the preprocessing stage. Different bandwidth values for the SLM were tested to find the optimal bandwidth W . However, the use of SLM did not improve the accuracy respect to the systematic selection of all the spectral peaks (see Fig. 7.11). This can be partially explained because in the test set there were only harmonic components, making unnecessary to discard spurious peaks. Besides this reason, the SLM method assumes that there are not two sinusoidal components closer than W . In some cases, this assumption does not hold in polyphonic real signals, where typical values of $W \in [10, 50]$ Hz exceed some partial frequency differences. Therefore, the SLM stage was removed to obtain the results described in Sec. 7.4.2, and the spectral peaks have been

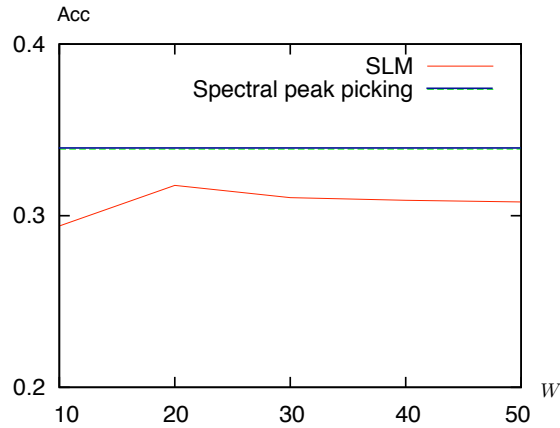


Figure 7.11: SLM accuracy respect to the bandwidth W using $\tau = 0.1$, and comparison with simple spectral peak picking. The other parameters used for the evaluation are those described in Tab. 7.1.

systematically selected from the magnitude spectrum instead, as SML did not improve neither the accuracy nor the efficiency with the tested values.

Experimentally, the use of all the spectral peaks yielded exactly the same results than the selection of those spectral peaks with a magnitude over a low fixed threshold $\mu = 0.1$. This thresholding, which did not alter the results in the test set, can reduce the computation time of the overall system to the half²¹. For this reason, this threshold was adopted and subsequently included in the joint estimation methods.

The overall results without SLM can be seen in Fig. 7.12. In this figure, the chosen parameter values are in the central intersection, and they correspond to those described in Tab. 7.1. From these initial values, the parameters have been changed individually to compare their impact in the accuracy.

The zero padding factor z is useful to accurately identify the frequency of lower pitches. As shown in Fig. 7.12, the overall accuracy increases importantly when using zero padding²² ($z \neq 2^0$). The computational cost derived from the FFT computation of longer windows must also be taken into account. As the overall computational cost of this method is very low, a value $z = 8$, which slightly improves the accuracy, was chosen.

The range of valid fundamental frequencies comprises the f_0 range of the data set used for the evaluation, therefore it is the same for all the evaluated methods.

The closest pitch distance value matches the spectral resolution obtained with zero padding. This way, using a margin of $f_d = 3$ Hz, only spectral

²¹Experimentally, the running time was reduced from 146.05 to 73.4 seconds.

²²Due to the FFT constraints, only power of 2 values for z have been tested.

7. MULTIPLE F_0 ESTIMATION USING SIGNAL PROCESSING METHODS

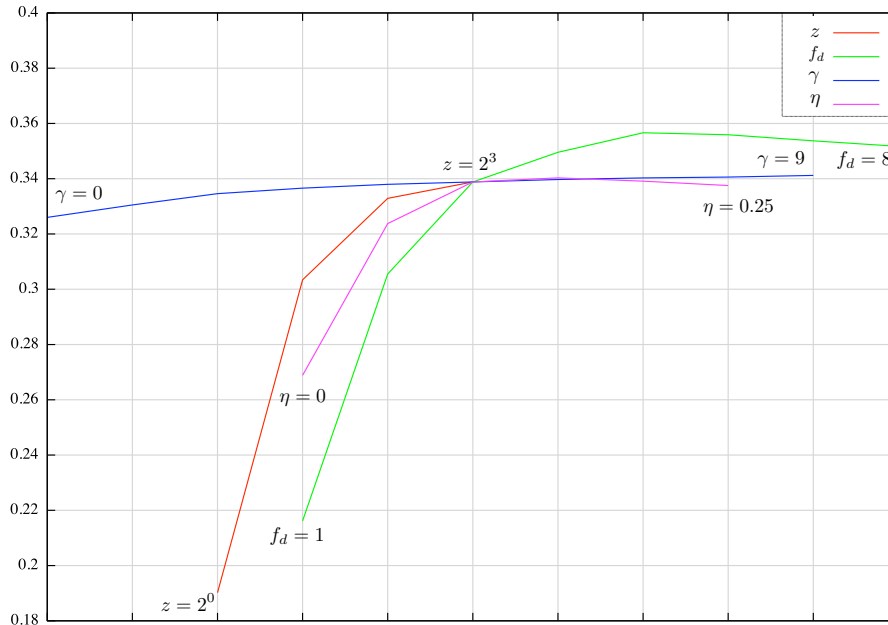


Figure 7.12: Iterative cancellation accuracy adjusting the free parameters. The abscissae axis is not labeled since these values depend on each particular parameter (see the first and last values for each parameter to get the grid step).

peaks at ± 1 bin from the ideal pitch frequency are considered as f_0 candidates. This parameter increases the accuracy about a 1%. However, as it can be seen in Fig. 7.12, the value selected for this parameter ($f_d = 3$) is probably too much restrictive, and a higher range ($f_d = 5$) yields better results. It must be considered that the iterative cancellation method was developed before having the random mixtures database, therefore their parameters are not optimally tuned for this data set. However, experimentally, the accuracy deviation shows that the chosen values do not differ much from the ones that approximate to the highest accuracy using this data set.

The postprocessing parameters of the iterative cancellation method are the minimum note intensity γ and the minimum relative intensity η of a candidate respect to the maximum intensity of the other simultaneous candidates in the analyzed frame. The note silence threshold value $\gamma = 5$ (equivalent to 18.38 dB) could not be directly evaluated using the random mixtures data set, as there are no silences and all the sounds have very similar amplitudes. However, the results varying $\gamma \in [0, 9]$ show that this value has a low impact in the detection when there are no silent excerpts in the signal.

Stage	Parameter	Symbol	Value
Preprocessing	Partial selection threshold	μ	0.1
	Zero padding factor	z	4
Candidate selection	Min f_0 amplitude	ε	2
	f_0 range	$[f_{min}, f_{max}]$	[38, 2100] Hz
Combination generation	Max number of candidates	F	10
	Max polyphony	P	6
Saliency calculation	Partial search bandwidth	f_r	11 Hz
	HPS length	H	10
	Absolute intensity threshold	γ	5
	Relative intensity threshold	η	0.1
	Smoothness weight	κ	2
Postprocessing	Minimum note duration	d	55.68 ms

Table 7.2: Parameters for the joint estimation method I.

The parameter $\eta = 0.1$ assumes that the relative intensity of a note is as minimum a 10% respect to the maximum intensity in the mixture²³. The effect of this parameter in the detection is shown in Fig. 7.12. It has been considered that all the notes in the database have similar amplitudes, therefore this threshold should have a low value for real signals out of the data set, which are usually of different intensities.

Joint estimation method I

The parameters used for the joint estimation I method are shown in Tab. 7.2, and the impact in the accuracy when they vary can be seen in Fig. 7.14.

A peak picking threshold $\mu = 0.1$ was chosen, like in the iterative cancellation method. This parameter increased the efficiency with a very low accuracy cost²⁴.

Like in the iterative cancellation method, the zero padding factor has shown to be very relevant to increase the accuracy (see Fig. 7.14). A trade-off value $z = 4$, instead of $z = 8$ used in iterative cancellation method, was chosen to avoid increasing significantly the computational cost (see Fig. 7.15), which is higher in this method.

The minimum f_0 bin amplitude $\varepsilon = 2$ slightly increases the accuracy and decreases the candidate selection error (see Fig. 7.13). A higher accuracy was obtained with $\varepsilon = 5$, but note that this parameter must have a lower value for the analysis of real musical signals²⁵, so this more conservative value was selected instead.

²³It is assumed that, with lower intensities, notes are usually masked by the amplitude of the other pitches in the mixture, and they can hardly be perceived.

²⁴Experimentally, in this method the accuracy decreased from 0.553 to 0.548.

²⁵Unlike in real music, in the test set all the signals had similar and high amplitudes.

7. MULTIPLE F_0 ESTIMATION USING SIGNAL PROCESSING METHODS

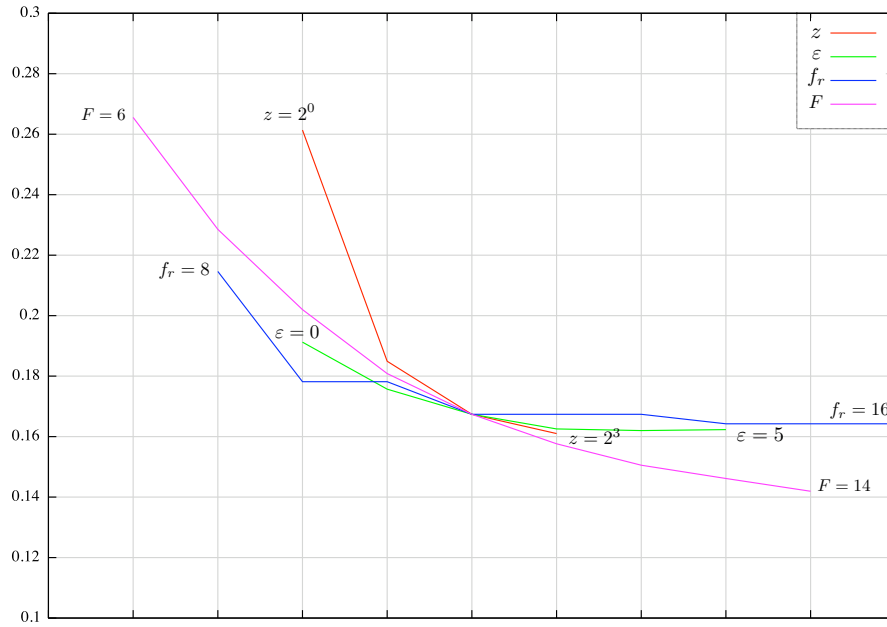


Figure 7.13: Joint estimation method I candidate error rate adjusting the parameters that have some influence in the candidate selection stage.

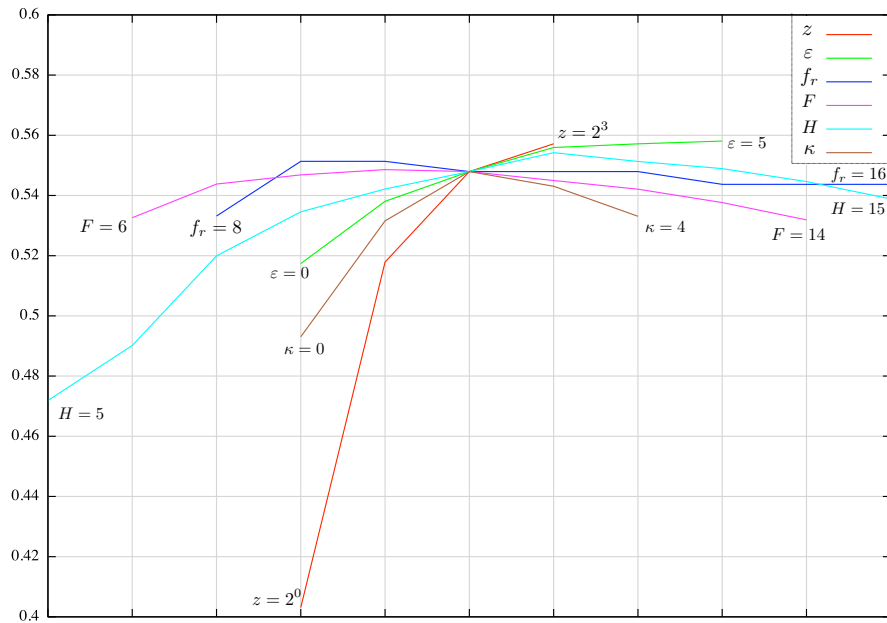


Figure 7.14: Joint estimation method I accuracy adjusting the free parameters.

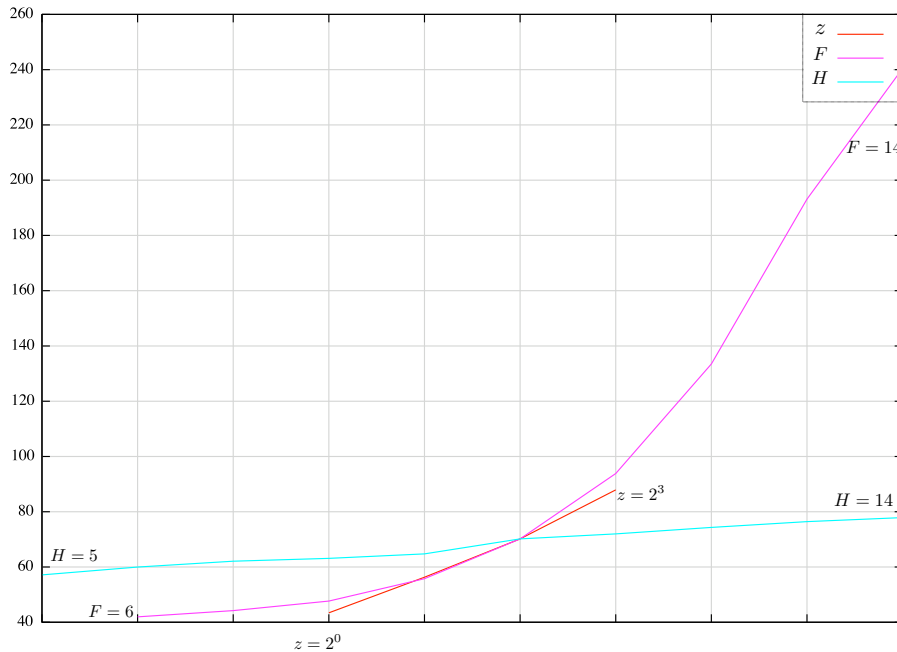


Figure 7.15: Joint estimation method I runtime in seconds adjusting the parameters that have some influence in the computational cost.

The bandwidth for searching partials f_r does not seem to have a great impact in the accuracy, but it is important in the candidate selection stage (see Fig. 7.13). An appropriate balance between a high accuracy and a low candidate selection error rate was obtained using $f_r = 11$ Hz.

The computational cost increases exponentially with the number of candidates F . Therefore, a good choice of F is critical for the efficiency of the method. Experimentally, $F = 10$ yielded a good trade-off between the accuracy, the number of correctly selected candidates and the computational cost (see Figs. 7.13, 7.14 and 7.15).

As previously mentioned, the first partials usually contain most of the energy of the harmonic sounds. Experimentally, using $H = 10$ suffices, and higher values cause low pitches to cancel other higher frequency components. In addition, note that the computational cost linearly increases with respect to H .

The smoothness weight which maximizes the accuracy was experimentally found using $\kappa = 2$. It is important to note that without considering spectral smoothing ($\kappa = 0$), the accuracy decreases significantly (see Fig. 7.14).

The postprocessing parameter values for γ and η were selected with the same values than in the iterative cancellation approach.

7. MULTIPLE F_0 ESTIMATION USING SIGNAL PROCESSING METHODS

Stage	Parameter	Symbol	Value
Preprocessing	Partial selection threshold	μ	0.1
	Zero padding factor	z	4
Candidate selection	f_0 range	$[f_{min}, f_{max}]$	[38, 2100] Hz
Combination generation	Max number of candidates	F	10
	Max polyphony	P	6
Saliency calculation	Partial search bandwidth	f_r	11 Hz
	HPS length	H	15
	Absolute intensity threshold	γ	5
	Relative intensity threshold	η	0.15
Postprocessing (without tracking)	Smoothness weight	κ	4
	Minimum note duration	d	23 ms
	Minimum rest duration	r	50 ms
Postprocessing (with tracking)	Number of adjacent frames	K	2

Table 7.3: Parameters for the joint estimation method II.

An additional experiment was done to measure the importance of the square factor in Eq. 7.13. Without squaring, the accuracy decreased from 0.548 to 0.476, which is significantly lower, showing the importance of this factor introduced to favor the sparseness.

Joint estimation method II

The parameters selected for the joint estimation method II are shown in Tab. 7.3. Most of them are the same than in the method I, except from H , η and κ , which yielded better results with slightly different values (see Fig. 7.16). In the case of H and κ , the values that maximized the accuracy were selected.

Like in the joint estimation method I, the parameter $\eta = 0.15$ has been set with a conservative value in order to avoid that the system performs worse for real musical signals, which usually do not have very similar intensities for the different sounds.

The postprocessing parameters can not be directly evaluated with this data set, as they have a temporal dimension and each mixture is composed of a single combination of pitches. However, a value $K = 2$ (considering 2 previous frames, 2 posterior frames and the target frame) has proven to be adequate for the analysis of real musical signals out from the data set. This value provides a notable temporal smoothness without altering significantly the temporal resolution required for short notes.

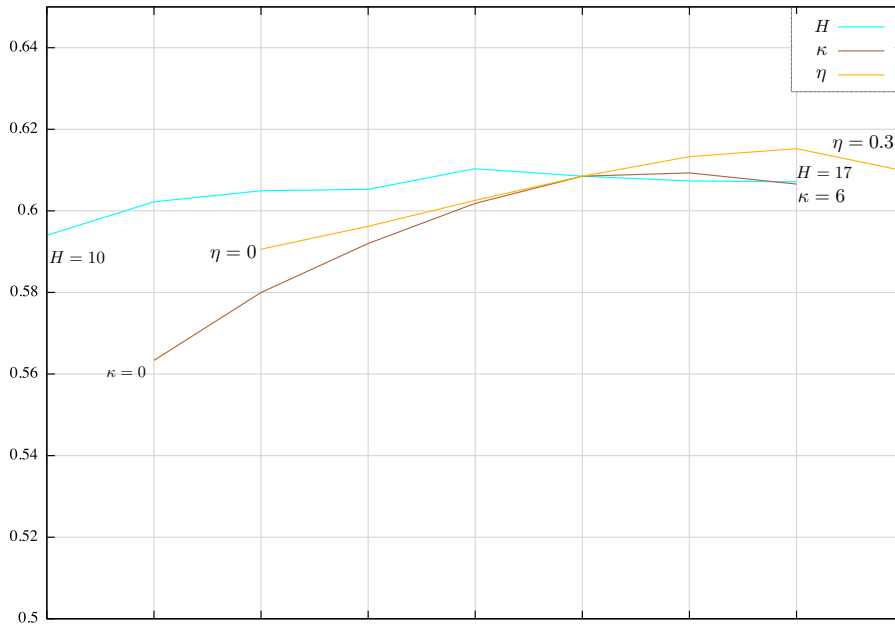


Figure 7.16: Joint estimation method II accuracy respect to the parameters that changed respect to the method I.

7.4.2 Results using random mixtures

The overall results for the random mixtures data set after the parametrization stage are described in the following figures of this section (Figs. 7.17 to 7.30). These results can not be directly compared to the evaluation made by Klapuri (2006b) using the same data, as in this latter work polyphony estimation and f_0 estimation were evaluated separately (the number of concurrent sounds was given as a parameter for the pitch estimator), whereas in the present work these two stages are calculated simultaneously.

As shown in Figs. 7.17 and 7.18, the candidate identification technique used in the joint estimation method II outperforms the other candidate selection approaches. It can also be seen (Figs. 7.19 to 7.22) that the joint estimation method I clearly outperforms the iterative cancellation approach, and the joint estimation method II gets a higher accuracy than the joint method I.

Respect to the estimation of the number of concurrent sources (Figs. 7.23 to 7.27), the joint estimation method II usually yields better results, but when there are many simultaneous sources (Fig. 7.26), it tends to underestimate the number of concurrent sounds, probably due to the combination of adjacent frames. Looking at the evaluation in function of the pitch (Figs. 7.28 to 7.30), it can be seen that the best results are located in the central pitch range.

7. MULTIPLE F_0 ESTIMATION USING SIGNAL PROCESSING METHODS

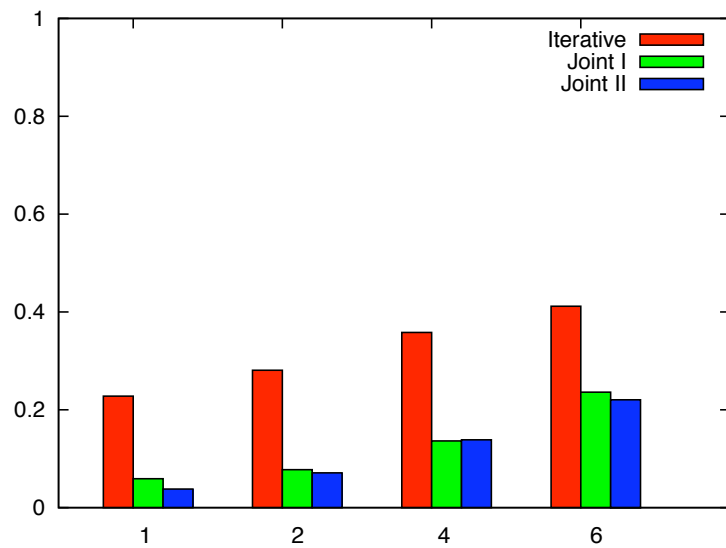


Figure 7.17: Candidate identification error rate with respect to the polyphony (1, 2, 4 and 6 simultaneous pitches) of the ground truth mixtures.

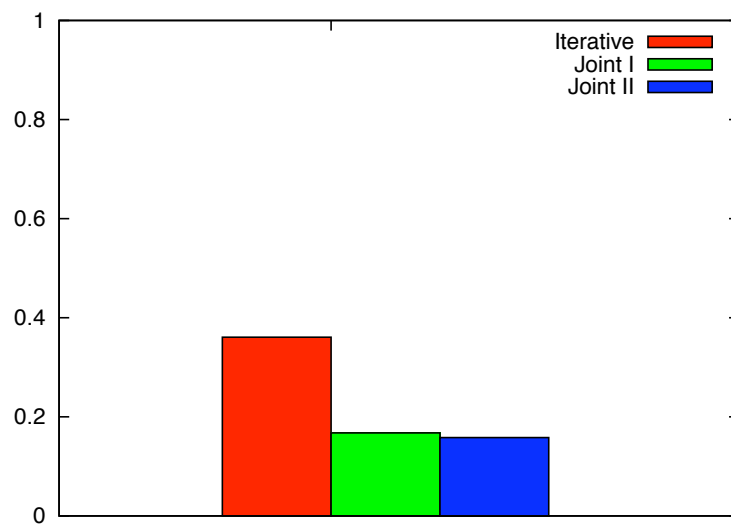


Figure 7.18: Global candidate identification error rate.

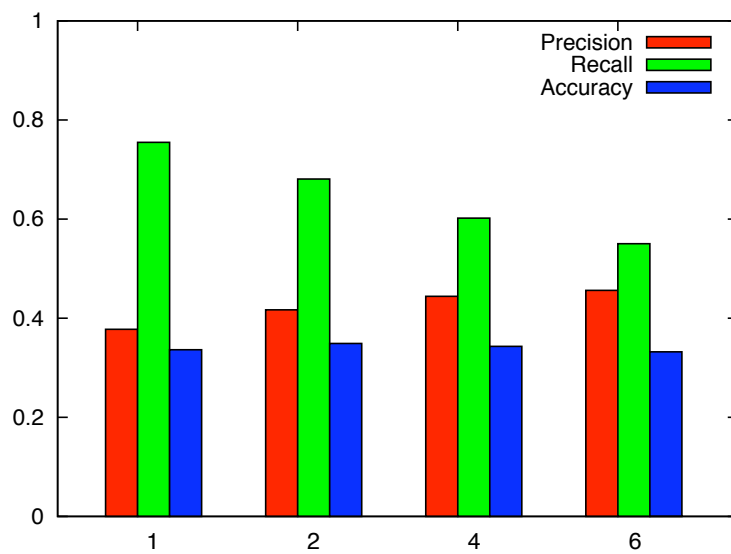


Figure 7.19: Pitch detection results for the iterative cancellation method with respect to the ground-truth mixtures polyphony.

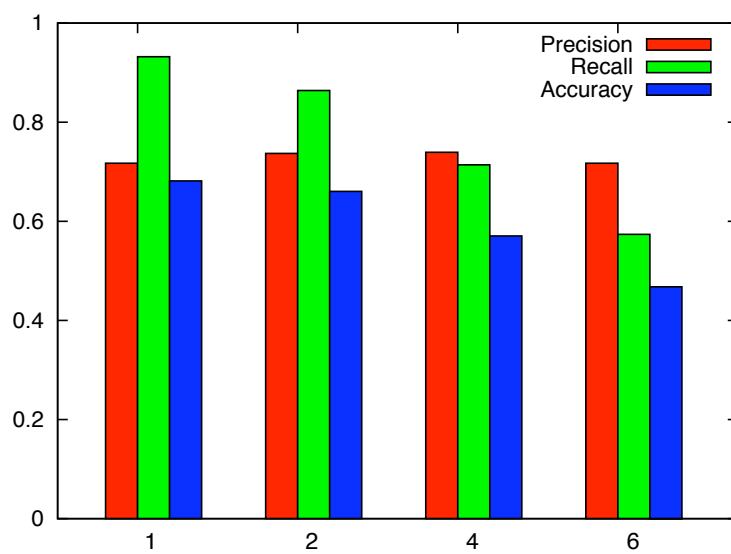


Figure 7.20: Pitch detection results for the joint estimation method I with respect to the ground-truth mixtures polyphony.

7. MULTIPLE F_0 ESTIMATION USING SIGNAL PROCESSING METHODS

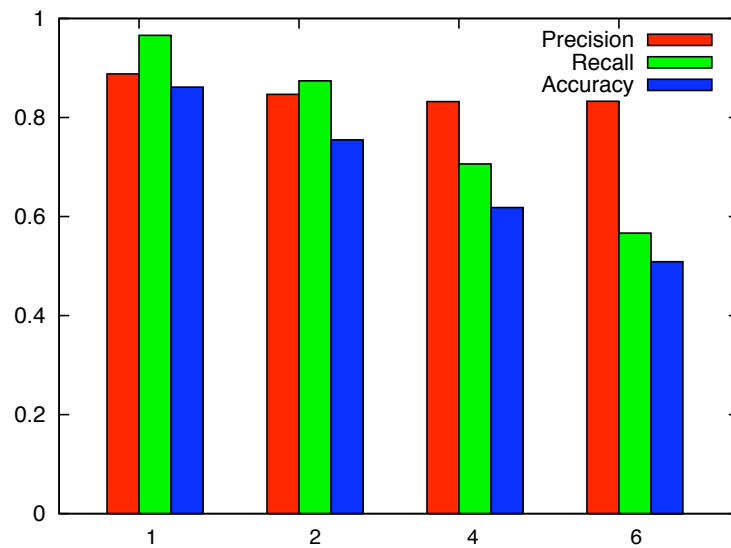


Figure 7.21: Pitch detection results for the joint estimation method II with respect to the ground-truth mixtures polyphony.

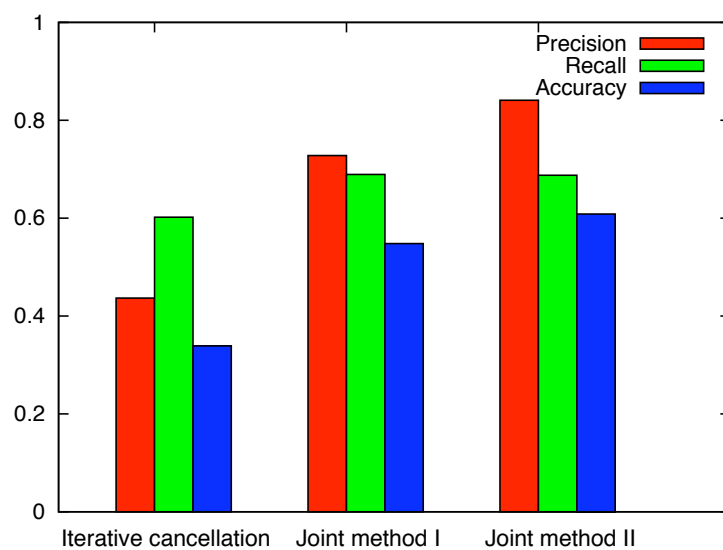


Figure 7.22: Comparison of the global pitch detection results for the three methods.

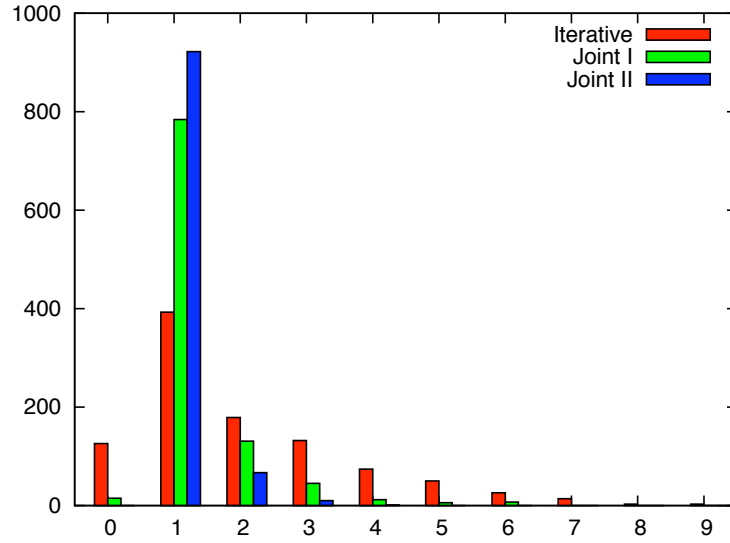


Figure 7.23: Number of concurrent sounds estimated for a single source.

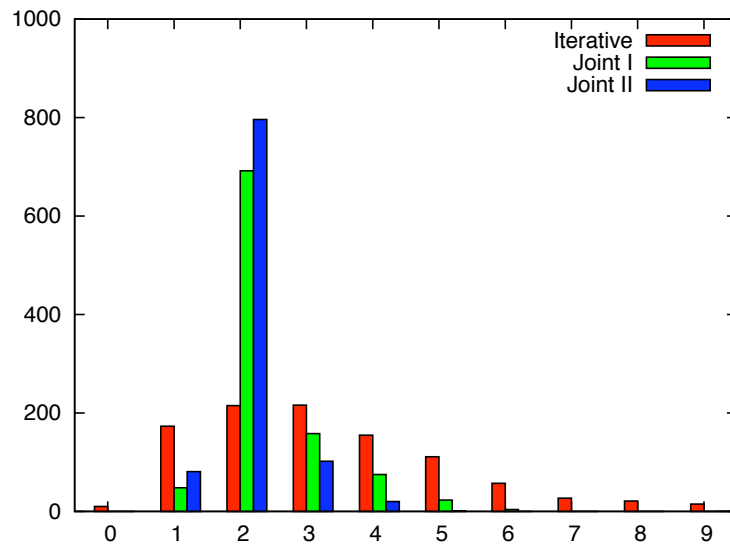


Figure 7.24: Number of concurrent sounds estimated for two simultaneous sources.

7. MULTIPLE F_0 ESTIMATION USING SIGNAL PROCESSING METHODS

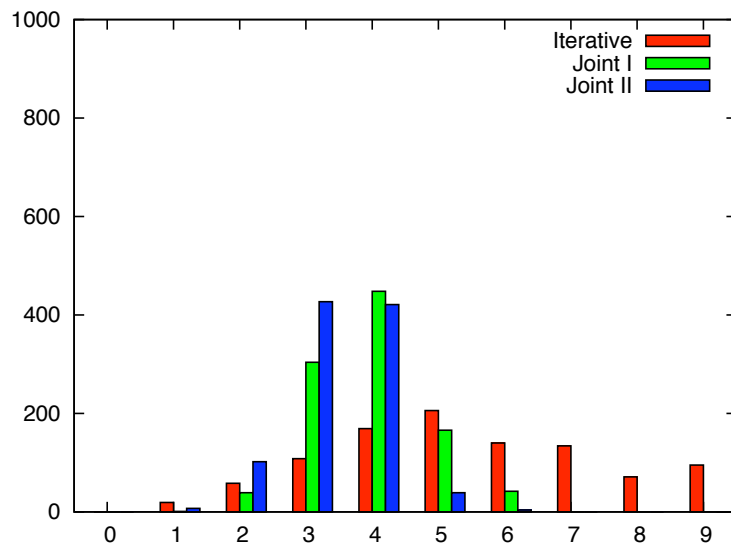


Figure 7.25: Number of concurrent sounds estimated for four simultaneous sources.

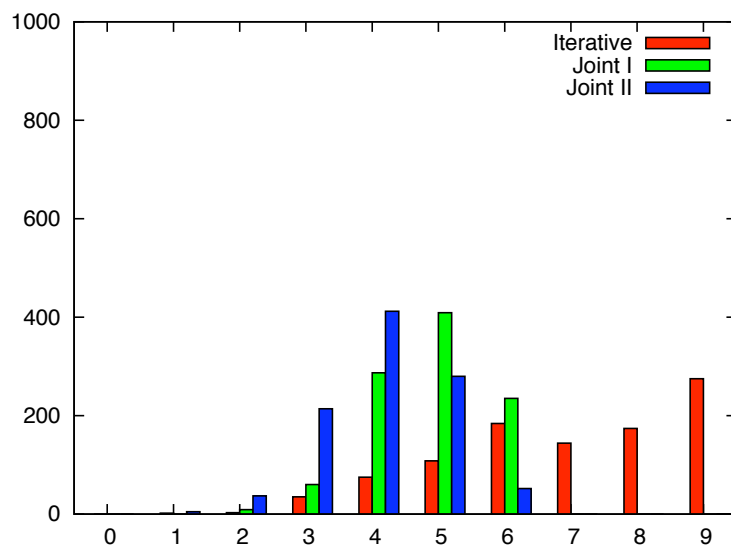


Figure 7.26: Estimation of the number of concurrent sources for six simultaneous sources.

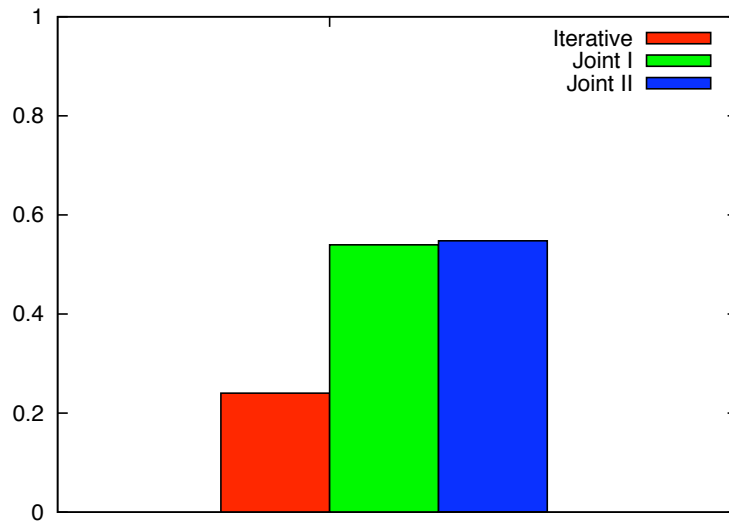


Figure 7.27: Global polyphony estimation results.

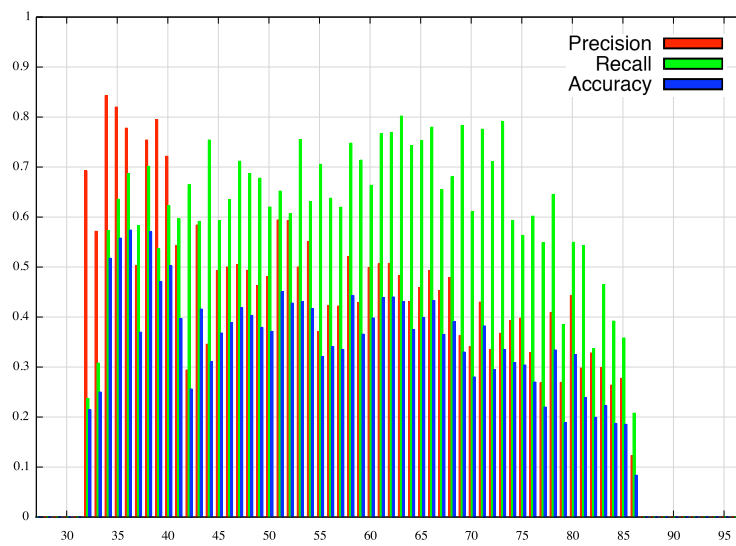


Figure 7.28: Precision, recall and accuracy of the iterative cancellation method in function of the MIDI pitch number.

7. MULTIPLE F_0 ESTIMATION USING SIGNAL PROCESSING METHODS

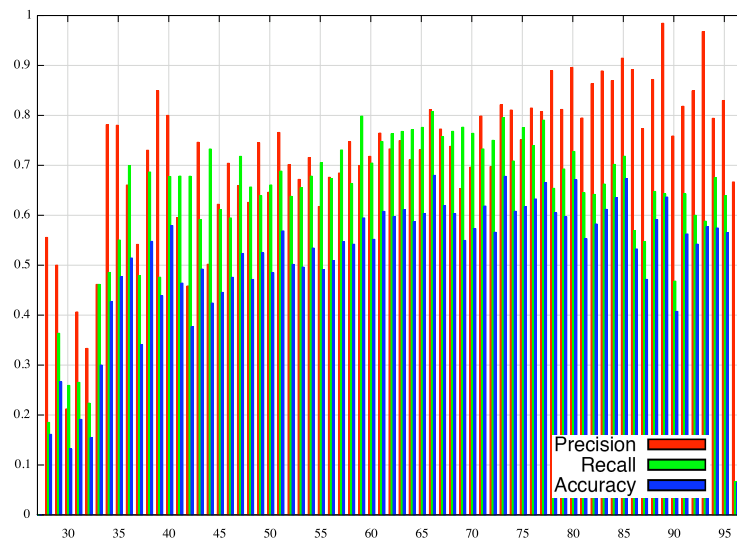


Figure 7.29: Precision, recall and accuracy of the joint estimation method I in function of the MIDI pitch number.

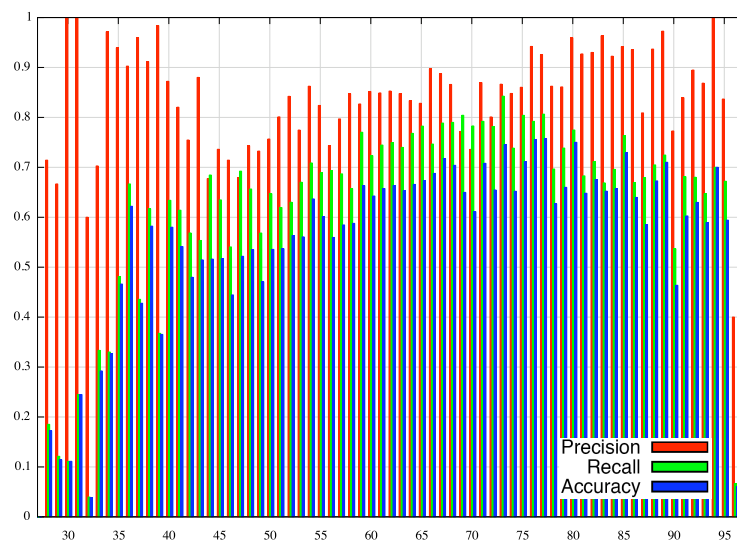


Figure 7.30: Precision, recall and accuracy of the joint estimation method II in function of the MIDI pitch number.

7.4.3 Evaluation and comparison with other methods

In order to evaluate the proposed methods using real musical signals and to compare them with other approaches, the iterative cancellation algorithm and the joint estimation method I were submitted to the [MIREX \(2007\)](#) multiple f_0 estimation and tracking contest, whereas the joint estimation method II was evaluated in [MIREX \(2008\)](#).

The data set used in [MIREX \(2007\)](#) and [MIREX \(2008\)](#) were essentially the same, consisting of a woodwind quintet transcription of the fifth variation from Beethoven, plus some synthesized pieces using [Goto \(2003\)](#) samples, and polyphonic piano recordings using a disklavier piano. There were clips of 30 seconds for each polyphony (2-3-4-5), for a total of 30 examples, plus 10 polyphonic piano pieces of 30 seconds. The details of the ground-truth labelling are described in ([Bay et al., 2009](#)).

The MIREX evaluation was done at two different levels; frame by frame pitch estimation and note tracking. The first mode evaluates the correct detection in isolated frames, whereas the second task also considers the temporal coherence of the detection.

For the frame level task, evaluation of the active pitches is done every 10 ms. For this reason, the hop size of the joint estimation methods²⁶ were set to obtain an adequate temporal resolution. Precision, recall, and accuracy were reported. A returned pitch is assumed to be correct if it is within a half semitone of a ground-truth pitch for that frame. Only one ground-truth pitch can be associated with each returned pitch. The error metrics from [Poliner and Ellis \(2007a\)](#) and previously described in pag. 51 were also used in the evaluation.

For the note tracking task, precision, recall, and F-measure were reported. A ground-truth note is assumed to be correctly transcribed if the method returns a note that is within a half semitone of that note, the yielded note onset is within a ± 50 ms range of the onset of the ground truth note, and its offset is within 20% range of the ground truth note offset. One ground truth note can only be associated with one transcribed note.

The data set is not publicly available, therefore the experiments using these data can not be replicated out of the MIREX contests.

Iterative cancellation method

The iterative cancellation approach does not perform a frame by frame evaluation, as it uses only those frames that are after the detected onsets to yield the pitches for each inter-onset interval. Although it does not perform f_0 tracking, onset times provide the indications about the beginning of the notes, therefore it was only submitted to the note tracking task.

²⁶The iterative cancellation method was only presented to the note tracking task.

7. MULTIPLE F_0 ESTIMATION USING SIGNAL PROCESSING METHODS

Participant	Runtime (sec)	Machine
Iterative cancellation	165	ALE Nodes
Joint estimation method I	364	ALE Nodes
AC3	900	MAC
AC4	900	MAC
EV4	2475	ALE Nodes
EV3	2535	ALE Nodes
KE3	4140	ALE Nodes
PE2	4890	ALE Nodes
RK	3285	SANDBOX
KE4	20700	ALE Nodes
VE	390600	ALE Nodes

Table 7.4: [MIREX \(2007\)](#) note tracking runtimes. Participant, running time (in seconds), and machine where the evaluation was performed are shown.

The parameters of the iterative cancellation method are those described in Tab. 7.1, using SLM in the preprocessing stage. Although the joint estimation method I is a pure frame by frame estimation approach, it was also submitted for the note tracking task, to compare the results with the iterative approach at least at note level.

The overall results using onset and pitch²⁷ are shown in Tab. 7.5, and the runtimes are in Tab. 7.4. The iterative cancellation method was the one with the lowest computational cost. As expected, the F-measure was not very high, but the method outperformed two complex approaches. However, joint estimation method I, which is really a frame by frame estimation method, obtained better results than the iterative cancellation approach for the note tracking task.

Joint estimation method I

The joint estimation method I was submitted for evaluation in [MIREX \(2007\)](#) frame by frame and note tracking contests with the parameters specified in Tab. 7.2.

The results for the frame by frame analysis are shown in Tab. 7.6, and the corresponding runtimes in Tab. 7.7. The accuracy of this method was close²⁸ to the highest accuracy among the evaluated methods, being the one with the highest precision and the lowest E_{tot} error. The precision, recall and accuracy were slightly better than those obtained with the random mixtures database. Probably, this is because the random mixtures database spans a pitch range wider than the MIREX data set, and very low or very high frequency pitches are those harder to detect (see Fig. 7.29).

²⁷Evaluation for onset, offset and pitch were also done in [MIREX \(2007\)](#), but results are not reported in this work, as the iterative estimation system does not consider offsets.

²⁸About 0.025 lower.

id	Participant	Method	Avg. F-m	Prec	Rec	Avg. Overlap
RK	Ryyänen and Klapuri (2005)	Iterative cancellation + HMM tracking	0.614	0.578	0.678	0.699
EV4	Vincent et al. (2007)	Unsupervised learning (NMF)	0.527	0.447	0.692	0.636
PE2	Poliner and Ellis (2007a)	Supervised learning (SVM)	0.485	0.533	0.485	0.740
EV3	Vincent et al. (2007)	Unsupervised learning (NMF)	0.453	0.412	0.554	0.622
PI2	Pertusa and Iñesta (2008a)	Joint estimation method I	0.408	0.371	0.474	0.665
KE4	Kameoka et al. (2007)	Statistical spectral models	0.268	0.263	0.301	0.557
KE3	Kameoka et al. (2007)	Statistical spectral models	0.246	0.216	0.323	0.610
PI3	Lidy et al. (2007)	Iterative cancellation	0.219	0.203	0.296	0.628
VE2	Emiya et al. (2007, 2008b)	Joint estimation + Bayesian models	0.202	0.338	0.171	0.486
AC4	Cont (2007)	Unsupervised learning (NMF)	0.093	0.070	0.172	0.536
AC3	Cont (2007)	Unsupervised learning (NMF)	0.087	0.067	0.137	0.523

Table 7.5: MIREX (2007) note tracking results based on onset and pitch. Average F-measure, precision, recall, and average overlap are shown for each participant.

7. MULTIPLE F_0 ESTIMATION USING SIGNAL PROCESSING METHODS

id	Participant	Method	Acc	Prec	Rec	E_{tot}	E_{subs}	E_{miss}	E_{fa}
RK	Ryynänen and Klapuri (2005)	Iterative cancellation + HMM tracking	0.605	0.690	0.709	0.474	0.158	0.133	0.183
CY	Yeh (2008)	Joint estimation	0.589	0.765	0.655	0.460	0.108	0.238	0.115
ZR	Zhou et al. (2009)	Saliency function (RTFI)	0.582	0.710	0.661	0.498	0.141	0.197	0.160
PH	Pertusa and Inesta (2008a)	Joint estimation method I	0.580	0.827	0.608	0.445	0.094	0.298	0.053
EV2	Vincent et al. (2007)	Unsupervised learning (NMF)	0.543	0.687	0.625	0.538	0.135	0.240	0.163
CC1	Cao et al. (2007)	Iterative cancellation	0.510	0.567	0.671	0.685	0.200	0.128	0.356
SR	Raczynski et al. (2007)	Unsupervised learning (NNMA)	0.484	0.614	0.595	0.670	0.185	0.219	0.265
EV1	Vincent et al. (2007)	Unsupervised learning (NMF)	0.466	0.659	0.513	0.594	0.171	0.371	0.107
PE1	Poliner and Ellis (2007a)	Supervised learning (SVM)	0.444	0.734	0.505	0.639	0.120	0.375	0.144
PL	Leveau (2007)	Matching pursuit	0.394	0.689	0.417	0.639	0.151	0.432	0.055
CC2	Cao et al. (2007)	Iterative cancellation	0.359	0.359	0.767	1.678	0.232	0.001	1.445
KE2	Kameoka et al. (2007)	Statistical spectral models (HTC)	0.336	0.348	0.546	1.188	0.401	0.052	0.734
KE1	Kameoka et al. (2007)	Statistical spectral models (HTC)	0.327	0.335	0.618	1.427	0.339	0.046	1.042
AC2	Cont (2007)	Unsupervised learning (NMF)	0.311	0.373	0.431	0.990	0.348	0.221	0.421
AC1	Cont (2007)	Unsupervised learning (NMF)	0.277	0.298	0.530	1.444	0.332	0.138	0.974
VE	Emiya et al. (2007, 2008b)	Joint estimation + Bayesian models	0.145	0.530	0.157	0.957	0.070	0.767	0.120

Table 7.6: MIREX (2007) frame by frame evaluation results. Accuracy, precision, recall, and the error metrics proposed by Poliner and Ellis (2007a) are shown for each participant.

id	Runtime (sec)	Machine
ZR	271	BLACK
Joint estimation method I	364	ALE Nodes
AC1	840	MAC
AC2	840	MAC
EV2	2233	ALE Nodes
EV1	2366	ALE Nodes
CC1	2513	ALE Nodes
CC2	2520	ALE Nodes
RK	3540	SANDBOX
PE1	4564	ALE Nodes
PL	14700	ALE Nodes
KE2	19320	ALE Nodes
KE1	38640	ALE Nodes
SR	41160	ALE Nodes
CY	132300	ALE Nodes
VE	364560	ALE Nodes

Table 7.7: MIREX (2007) frame by frame runtimes. The first column shows the participant, the second is the runtime and the third column is the machine where the evaluation was performed. ALE Nodes was the fastest machine.

The method was also evaluated in the note tracking contest. Despite it was not designed for this task, as the analysis is performed without information of neighboring frames but converting consecutive pitch detections into notes, the results were not bad, as shown in Tab. 7.5.

The joint estimation method I was also very efficient respect to the other state of the art methods presented (see Tab. 7.7), specially considering that it is a joint estimation approach.

Joint estimation method II

The joint estimation method II was submitted to MIREX (2008) for frame by frame and note tracking evaluation. The method was presented for both tasks in two setups: with and without f_0 tracking.

The difference between using f_0 tracking or not is the postprocessing stage (see Tab. 7.3). In the first setup, notes shorter than a minimum duration are just removed, and when there are short rests between two consecutive notes of the same pitch, the notes are merged. Using f_0 tracking, the methodology described in Sec. 7.3.3 is performed instead, increasing the temporal coherence of the estimate with the wDAG.

Experimentally, the joint estimation method II was very efficient compared to the other approaches presented, as shown in Tabs. 7.8 and 7.9.

The results for the frame by frame task can be seen in Tab. 7.10. The accuracy for the joint estimation method II without f_0 tracking is satisfactory,

7. MULTIPLE F_0 ESTIMATION USING SIGNAL PROCESSING METHODS

Participant	Runtime (sec)
MG	99
Joint estimation II	792
Joint estimation II + tracking	955
VBB	2081
CL1	2430
CL2	2475
RK	5058
EOS	9328
DRD	14502
EBD1	18180
EBD2	22270
YRC1	57483
YRC2	57483
RFF2	70041
RFF1	73784

Table 7.8: [MIREX \(2008\)](#) frame by frame runtimes. Participants and runtimes are shown. All the methods except MG were evaluated using the same machine.

Participant	Runtime (sec)
Joint estimation II	790
ZR3	871
Joint estimation II + tracking	950
ZR1	1415
ZR2	1415
VBB	2058
RK	5044
EOS	9328
EBD1	18180
EBD2	22270
YRC	57483
RFF2	71360
RFF1	73718

Table 7.9: [MIREX \(2008\)](#) note tracking runtimes. Participants and runtimes are shown. All the methods except ZR were evaluated using the same machine.

and the method obtained the highest precision and the lowest E_{tot} error among all the analyzed approaches.

The inclusion of f_0 tracking did not improve the results for frame by frame estimation, but in the note tracking task (see Tab. 7.11), the results outperformed those obtained without tracking.

7.4.4 Overall MIREX comparison

As the ground-truth used for [MIREX \(2007\)](#) and [MIREX \(2008\)](#) multiple f_0 estimation and tracking contest were the same. In the review from [Bay et al. \(2009\)](#), the results of the algorithms evaluated in both MIREX editions are analyzed.

id	Participant	Method	Acc	Prec	Rec	E_{tot}	E_{subs}	E_{miss}	E_{fa}
YRC2	Yeh et al. (2008)	Joint estimation + f_0 tracking	0.665	0.741	0.780	0.426	0.108	0.127	0.190
YRC1	Yeh et al. (2008)	Joint estimation	0.619	0.698	0.741	0.477	0.129	0.129	0.218
PI2	Pertusa and Inesta (2008b)	Joint estimation II	0.618	0.832	0.647	0.406	0.096	0.257	0.053
RK	Rynänen and Klapuri (2005)	Iterative cancellation + HMM tracking	0.613	0.698	0.719	0.464	0.151	0.130	0.183
PI1	Pertusa and Inesta (2008b)	Joint estimation II + tracking	0.596	0.824	0.625	0.429	0.101	0.275	0.053
VBB	Vincent et al. (2007)	Unsupervised learning (NMF)	0.540	0.714	0.615	0.544	0.118	0.267	0.159
DRD	Durrieu et al. (2008)	Iterative cancellation	0.495	0.541	0.660	0.731	0.245	0.096	0.391
CL2	Cao and Li (2008)	Iterative cancellation	0.487	0.671	0.560	0.598	0.148	0.292	0.158
EOS	Egashira et al. (2008)	Statistical spectral models (HTC)	0.467	0.591	0.546	0.649	0.210	0.244	0.194
EBD2	Emiya et al. (2008a)	Joint estimation + Bayesian models	0.452	0.713	0.493	0.599	0.146	0.362	0.092
EBD1	Emiya et al. (2008a)	Joint estimation + Bayesian models	0.447	0.674	0.498	0.629	0.161	0.341	0.127
MG	Groble (2008)	Database matching	0.427	0.481	0.570	0.816	0.298	0.133	0.385
CL1	Cao and Li (2008)	Iterative cancellation	0.358	0.358	0.763	1.680	0.236	0.001	1.443
RFF1	Reis et al. (2008a)	Supervised learning (genetic)	0.211	0.506	0.226	0.854	0.183	0.601	0.071
RFF2	Reis et al. (2008a)	Supervised learning (genetic)	0.183	0.509	0.191	0.857	0.155	0.656	0.047

Table 7.10: MIREX (2008) frame by frame evaluation results. Accuracy, precision, recall, and the error metrics proposed by Poliner and Ellis (2007a) are shown for each method.

id	Participant	Method	Avg. F-m	Prec	Rec	Avg. Overlap
YRC	Yeh et al. (2008)	Joint estimation + f_0 tracking	0.355	0.307	0.442	0.890
RK	Ryynänen and Klapuri (2005)	Iterative cancellation + HMM tracking	0.337	0.312	0.382	0.884
ZR3	Zhou and Reiss (2008)	Saliency function (RTF1)	0.278	0.256	0.314	0.874
ZR2	Zhou and Reiss (2008)	Saliency function (RTF1)	0.263	0.236	0.306	0.874
ZR1	Zhou and Reiss (2008)	Saliency function (RTF1)	0.261	0.233	0.303	0.875
PI1	Pertusa and Inesta (2008b)	Joint estimation II + tracking	0.247	0.201	0.333	0.862
EOS	Egashira et al. (2008)	Statistical spectral models (HTC)	0.236	0.228	0.255	0.856
VBB	Vincent et al. (2007)	Unsupervised learning (NMF)	0.197	0.162	0.268	0.829
PI2	Pertusa and Inesta (2008b)	Joint estimation II	0.192	0.145	0.301	0.854
EBD1	Emiya et al. (2008a)	Joint estimation + Bayesian models	0.176	0.165	0.200	0.865
EBD2	Emiya et al. (2008a)	Joint estimation + Bayesian models	0.158	0.153	0.178	0.845
RFF2	Reis et al. (2008a)	Supervised learning (genetic)	0.032	0.037	0.030	0.645
RFF1	Reis et al. (2008a)	Supervised learning (genetic)	0.028	0.034	0.025	0.683

Table 7.11: MIREX (2008) note tracking results based on onset, offset, and pitch. Average F-measure, precision, recall, and average overlap are shown for each method.

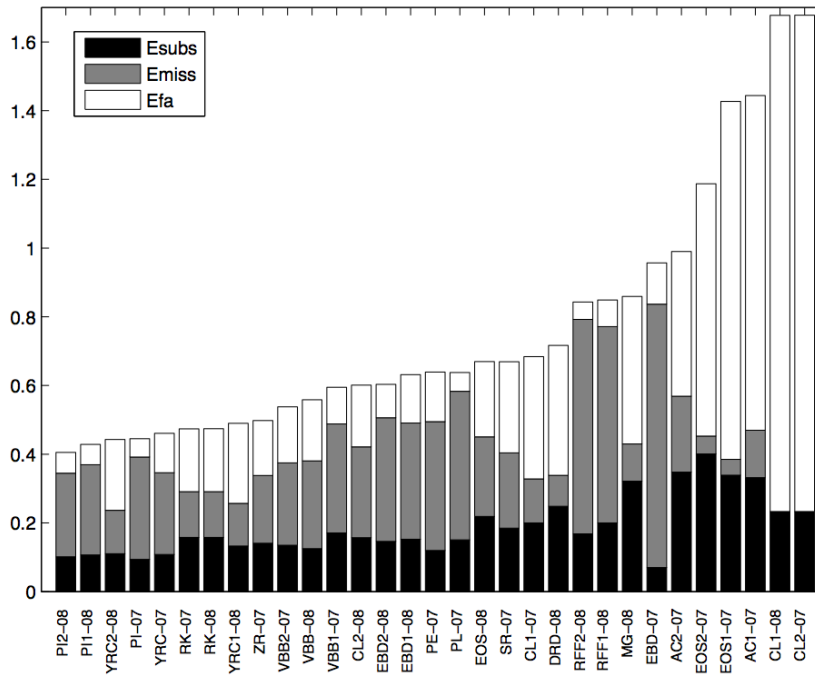


Figure 7.31: Fig. from Bay et al. (2009), showing E_{subs} , E_{miss} and E_{fa} for all MIREX 2007 and MIREX 2008 multiple fundamental frequency estimation methods ordered by E_{tot} . PI2-08 is the joint estimation method II without tracking, PI1-08 is the same method with tracking, and PI-07 is the joint estimation method I.

As shown in Fig. 7.32, the proposed joint estimation methods achieve a high overall accuracy and the highest precision rates among all the reference methods. The joint estimation method II also obtained the lowest error (E_{tot}) rate from the 31 methods submitted in both editions (see Fig. 7.31) using the metric proposed by Poliner and Ellis (2007a).

In the evaluation of note tracking considering only onsets, the proposed methods showed lower accuracies (Fig. 7.33), as only the joint estimation method II can perform a very basic f_0 tracking. With respect to the iterative cancellation approach, the accuracy was lower than the joint estimation methods, but not disappointing when compared to the other algorithms evaluated, given that this is a very simple method mainly intended for piano music.

As pointed out by Bay et al. (2009), the reason behind the different results in E_{tot} and accuracy in the frame by frame evaluation (see Figs. 7.31 and 7.32) is

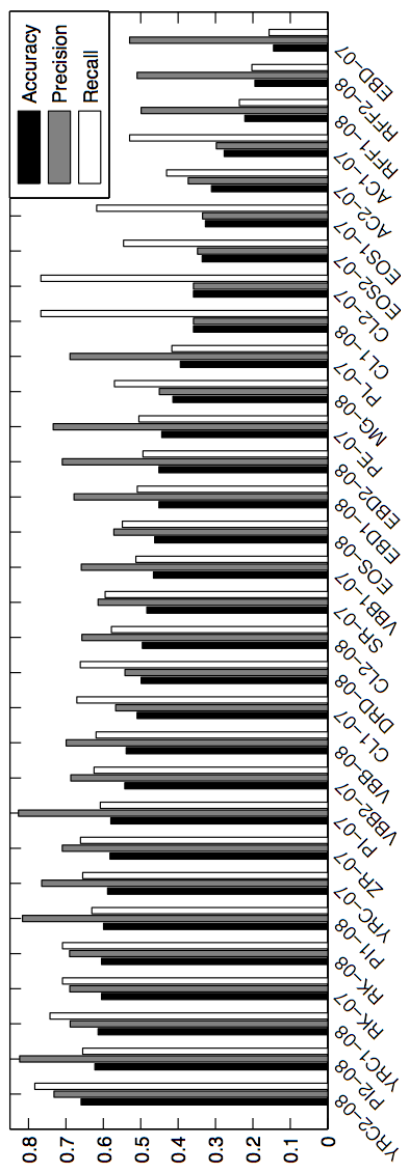


Figure 7.32: Fig. from Bay et al. (2009). Precision, recall and overall accuracy for all MIREX 2007 and MIREX 2008 multiple fundamental frequency estimation methods ordered by accuracy. PI2-08 is the joint estimation method II without tracking, PI1-08 is the same method with tracking, and PI-07 is the joint estimation method I.

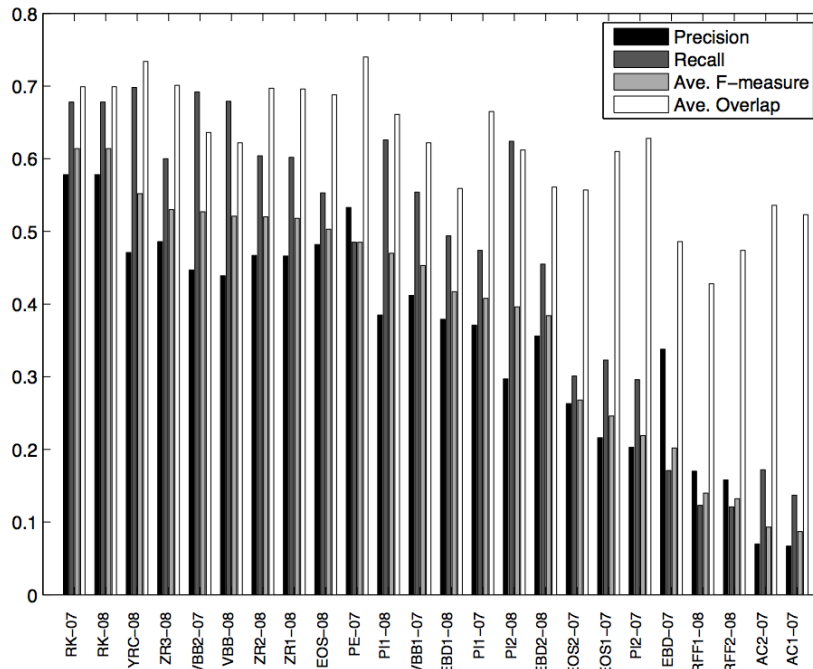


Figure 7.33: Fig. from Bay et al. (2009). Precision, recall, average F-measure and average overlap based on note onset for MIREX 2007 and MIREX 2008 note tracking subtask. PI2-08 is the joint estimation method II without tracking, PI1-08 is the same method with tracking, PI1-07 is the joint estimation method I and PI2-07 is the iterative cancellation method.

that most of the reported f_0 were correct, but multiple f_0 estimation algorithms tend to under-report and miss many active f_0 in the ground-truth.

While the proposed joint estimation methods I and II achieved the lowest E_{tot} score, there are very few false alarms compared to miss errors. On the other hand, the methods from Ryyänen and Klapuri (2005) and Yeh et al. (2008) have a better balanced precision, recall, as well as a good balance in the three error types, and as a result, have the highest accuracies for MIREX (2007) and MIREX (2008), respectively.

Citing Bay et al. (2009), "Inspecting the methods used and their performances, we can not make generalized claims as to what type of approach works best. In fact, statistical significance testing showed that the top three methods²⁹ were not significantly different."

²⁹(Yeh et al., 2008; Pertusa and Iñesta, 2008b; Ryyänen and Klapuri, 2005).

7.5 Conclusions

In this chapter, three different signal processing methods have been proposed for multiple f_0 estimation. Unlike the supervised learning approaches previously described, these signal processing schemes can be used to transcribe real music without any a-priori knowledge of the sources.

The first method is based on iterative cancellation, and it is a simple approach which is mainly intended for the transcription of piano sounds at a low computational cost. For this reason, only one frame in an inter-onset interval is analyzed, and the interaction between harmonic sources is not considered. A fixed spectral pattern is used to subtract the harmonic components of each candidate.

The joint estimation method I introduces a more complex methodology. The spectral patterns are inferred from the analysis of different hypotheses taking into account the interactions with the other sounds. The combination of harmonic patterns that maximizes a criterion based on the sum of harmonic amplitudes and spectral envelope smoothness is chosen at each frame.

The third method extends the previous joint estimation method considering adjacent frames for adding temporal smoothing. This method can be complemented with a f_0 tracking stage, using a weighted direct acyclic graph, to increase the temporal coherence of the detection.

The proposed methods have been evaluated and compared to other works. The iterative cancellation approach, mainly intended for piano transcription, is very efficient and it has been successfully used for genre classification and other MIR tasks (Lidy et al., 2007) with computational cost restrictions.

The joint estimation methods obtained a high accuracy and the lowest E_{tot} among all the multiple f_0 algorithms submitted in MIREX (2007) and MIREX (2008). Although all possible combinations of candidates are evaluated at each frame, the proposed approaches have a very low computational cost, showing that it is possible to make an efficient joint estimation method.

Probably, the f_0 tracking stage added to the joint estimation method II is too simple, and it should be replaced by a more reliable method in a future work. For instance, the transition weights could be learned from a labeled test set, or a more complex f_0 tracking method like the high-order HMM scheme from Chang et al. (2008) could be used instead. Besides intensity, the centroid of an HPS should also have a temporal coherence when belonging to the same source, therefore this parameter could also be considered for tracking.

Using stochastic models, a probability can be assigned to each pitch in order to remove those that are less probable given their context. For example, in a melodic line it is very unlikely that a non-diatonic note two octaves higher or lower than its neighbours appears. Musical probabilities can be taken into

account, like in (Ryynänen and Klapuri, 2005), to remove very improbable notes. The adaptation to polyphonic music of the stochastic approach from Pérez-Sancho (2009) is also planned as future work, in order to use it in the multiple f_0 estimation methods to obtain a musically coherent detection.

The evaluation and further research of the alternative architectures proposed for the joint estimation method II (see Sec. 7.3.4) is also left for future work.

8

Conclusions and future work

This work has addressed the automatic music transcription problem using different strategies. Efficient novel methods have been proposed for onset detection and multiple f_0 estimation, using supervised learning and signal processing techniques. The main contributions of this work can be summarized in the following points:

- An extensive review of the state of the art methods for onset detection and multiple f_0 estimation. The latter methods have been classified into salience functions, iterative cancellation, joint estimation, supervised learning, unsupervised learning, matching pursuit, Bayesian models, statistical spectral models, blackboard systems, and database matching methods. An analysis of the strengths and limitations for each category has also been done.
- The development of an efficient approach for onset detection and the construction of a ground-truth data set for this task. The main novelties in this field are the use of a 1/12 octave filter bank to compress the harmonic information and the simple onset detection functions proposed. The presented method is mainly intended for percussive onset detection, as it detects energy abrupt energy changes, but it also considers the properties of harmonic sounds, making it robust against spectral variations produced during the sustain stage of the sounds. The algorithm was evaluated and compared to other works yielding promising results.
- Two novel approaches for multiple pitch estimation of a priori known sounds using supervised learning methods. These algorithms were one of the first machine learning methods proposed for this task. A harmonic filter bank was used to reduce the amount of spectral information to feed a time-delay neural network (TDNN), while preserving the main harmonic content. A ground-truth data set of synthetic sounds was generated to

8. CONCLUSIONS AND FUTURE WORK

evaluate the method. The conclusions extracted from the comparison between the k nearest neighbors (k NN) and the time-delay neural networks for this task are also interesting. The TDNN clearly outperformed the results obtained by the k NN using synthesized mixtures, showing the advantages of the network for generalization within a large observable space. Alternative activation functions to generalize the k NN prototypes were also proposed, but the results were still far from those obtained with the TDNN.

- A simple iterative cancellation approach, mainly intended to transcribe piano music at a low computational cost. A complete system to load an audio file and generate a MIDI file was developed. A novel scheme, based on the analysis of the spectra only after each onset to improve the efficiency, was proposed, and a sinusoidal likeness measure was also investigated for this task. This method was the basis for the subsequent joint estimation approaches, and it has been successfully used for genre classification and other MIR tasks.
- Heuristic multiple f_0 algorithms based on signal processing to analyze real music without any a priori knowledge. These methods, which are probably the main contribution of this thesis, experimentally reached the state of the art for this task with a high efficiency. The harmonic patterns are inferred from the spectrum considering intensity and a novel smoothness measure in a joint scheme which takes into account the source interactions. The proposed approach also introduced a novel temporal smoothing technique by considering the pitch combinations in adjacent frames. A simple f_0 tracking method is also introduced using a weighted directed acyclic graph. These novel methods achieved high success rates, but also with a very high efficiency, which is the main handicap of joint estimation approaches.

8.1 Discussion and future lines of work

The proposed TDNN scheme could easily be adapted for onset detection in a future work. Like in the machine learning methods described for multiple pitch estimation, the 1/12 octave filter bank can be used to obtain the network input data, but for this application only an output neuron should be necessary to classify each frame as onset or not-onset. This learning scheme also implies to extend the compiled onset detection data set in order to get a larger and reliable training set.

The supervised learning method should also be trained and evaluated using real mixtures of different timbres and real audio signals aligned with the ground-truth pitches. As previously discussed, it is not an easy task to get an aligned

database. However, a similar scheme that proposed by (Yeh (2008)) could be used to build the training set. In the latter work, MIDI files are splitted into several files containing tracks of separate notes. Then, they are individually synthesized and the f_0 are estimated using the YIN algorithm from de Cheveigné and Kawahara (2002). The f_0 of the individual note samples collectively establish the ground truth of the synthesized polyphonic signal.

Probably, multiple f_0 estimation methods that only consider individual frames can hardly outperform the current approaches. The lack of data within that short period is not enough to detect the pitches even for an expert musician. Context plays an important role in music. For instance, it is very hard to detect the pitches when listening two songs at different tempo playing simultaneously, even when they are not very complex.

Therefore, pitch estimation should be complemented with temporal information in some way. The coherence of the detections along time has been considered in the joint estimation method II, but it could be extended using a reliable f_0 tracking method. However, as pointed out by Yeh (2008), f_0 tracking in a joint manner is complicated. A robust f_0 tracking algorithm should analyze many possible simultaneous pitch combinations at each frame for a long term, and this is a challenging task from a computational point of view. The problem can be simplified by tracking individual candidate trajectories like in the high-order HMM method from Chang et al. (2008).

The further research of the alternative schemes proposed for the joint estimation method II is also a promising research subject. The methodology of this approach allows, for instance, to merge combinations of those frames that are between two consecutive onsets, yielding the pitches within the inter-onset interval. Perceptually, the results obtained with this scheme were better than in the analysis of a few adjacent frames. However, the lower temporal resolution and the errors of the onset detection method, and the problem with the offsets in the inter-onset interval, condition the success rate using a classical frame by frame evaluation metric. As future work, it is planned to evaluate these schemes using a perceptual metric.

Multimodal information is also important for future research subjects. The inclusion of musical models by considering tonality, tempo, or meter to infer pitch probabilities could complement the pitch estimates. These lines of work are planned within the DRIMS project in collaboration with the Music Technology Group from the Universitat Pompeu Fabra.

Interactive music transcription is also planned as future work within the MIPRCV (Consolider Ingenio 2010) project. The main goal is to develop a computer-assisted method for music transcription, similarly to the system from Vidal et al. (2006) for machine translation. Using a visual interface, the portions of automatically transcribed music can be accepted or amended by an expert

musician. Then, these user-validated portions can be used by the multiple f_0 estimation method to produce further, hopefully improved suggestions by correcting frequent errors.

Further research of the proposed methods applied to other MIR tasks like genre or mood classification is also currently investigated in collaboration with the Department of Software Technology and Interactive Systems from the Vienna University of Technology.

8.2 Publications

Some contents of this thesis have been published in journals and conference proceedings. Here is a list of publications in chronological order.

- Pertusa, A. and Iñesta, J. M. (2004). Pattern recognition algorithms for polyphonic music transcription. In Fred, A., editor, *Pattern Recognition in Information Systems (PRIS)*, pages 80-89, Porto, Portugal. [Chapter 6]
- Pertusa, A., Klapuri, A., and Iñesta, J. M. (2005). Recognition of note onsets in digital music using semitone bands. *Lecture Notes in Computer Science*, 3773:869-879. [Chapter 5]
- Pertusa, A. and Iñesta, J. M. (2005). Polyphonic monotimbral music transcription using dynamic networks. *Pattern Recognition Letters*, 26(12):1809-1818. [Chapter 6]
- Lidy, T., Rauber, A., Pertusa, A., and Iñesta, J. M. (2007). Improving genre classification by combination of audio and symbolic descriptors using a transcription system. In *Proc. of the 8th International Conference on Music Information Retrieval (ISMIR)*, pages 61-66, Vienna, Austria. [Chapter 7]
- Pertusa, A. and Iñesta, J. M. (2007). Multiple fundamental frequency estimation based on spectral pattern loudness and smoothness. In MIREX (2007), multiple f_0 estimation and tracking contest. [Chapter 7]
- Lidy, T., Rauber, A., Pertusa, A., Ponce de León, P. J., and Iñesta, J. M. (2008). Audio music classification using a combination of spectral, timbral, rhythmic, temporal and symbolic features. In MIREX (2008), audio genre classification contest, Philadelphia, PA. [Chapter 7]
- Pertusa, A. and Iñesta, J. M. (2008). Multiple fundamental frequency estimation using Gaussian smoothness and short context. In MIREX (2008), multiple f_0 estimation and tracking contest. [Chapter 7]

- Pertusa, A. and Iñesta, J. M. (2008). Multiple fundamental frequency estimation using Gaussian smoothness. In *Proc. of the IEEE Int. Conf. on Acoustics, Speech, and Signal Processing (ICASSP)*, pages 105-108, Las Vegas, NV. [**Chapter 7**]
- Lidy, T., Grecu, A., Rauber, A., Pertusa, A., Ponce de León, P. J., and Iñesta, J. M. (2009). A multi-feature multi-classifier ensemble approach for audio music classification. In MIREX (2009), audio genre classification contest, Kobe, Japan. [**Chapter 7**]
- Pertusa, A. and Iñesta, J. M. (2009). Note onset detection using one semitone filter-bank for MIREX 2009. In MIREX (2009), onset detection contest, Kobe, Japan. [**Chapter 5**]

Summary in Spanish required by
the regulations of the University of
Alicante.



Resumen

Agradecimientos

Antes de nada, me gustaría agradecer a todos los miembros del grupo de música por ordenador de la Universidad de Alicante por proporcionar una excelente atmósfera de trabajo. Especialmente, al coordinador del grupo y supervisor de este trabajo, José Manuel Iñesta. Su incansable espíritu científico proporciona un marco de trabajo excelente para inspirar las nuevas ideas que nos hacen crecer y avanzar continuamente. Este trabajo no hubiera sido posible sin su consejo y ayuda.

Escribir una tesis no es una tarea fácil sin la ayuda de mucha gente. Primero, me gustaría agradecer a toda la plantilla de nuestro Grupo de Reconocimiento de Formas e Inteligencia Artificial (GRFIA) y, en general, a todo el Departamento de Lenguajes y Sistemas Informáticos (DLSI) de la Universidad de Alicante. Mis estancias de investigación con el *Audio Research Group, Tampere University of Technology* (Tampere), *Music Technology Group* (MTG), *Universitat Pompeu Fabra* (Barcelona) y *Department of Software Technology and Interactive Systems, Vienna University of Technology* (Viena), también han contribuido notablemente a la realización de este trabajo. He crecido mucho, como científico y como persona, aprendiendo de los integrantes de estos centros de investigación.

También me gustaría agradecer a la gente que ha contribuido directamente a este trabajo. A Francisco Moreno, por retrasar algunas de mis responsabilidades docentes durante la escritura de este documento y por proporcionar el código de los algoritmos de k vecinos más cercanos. He aprendido la mayoría de las técnicas que conozco para transcripción musical de Anssi Klapuri. Estaré eternamente agradecido por los grandes momentos que pasé en Tampere y por su generosa acogida. Anssi ha contribuido directamente a esta tesis proporcionando las bases para el código de similitud sinusoidal y la base de datos de acordes aleatorios que han posibilitado la evaluación y la mejora de los algoritmos

A. RESUMEN

propuestos. En este trabajo también ha colaborado directamente uno de mis estudiantes de grado, Jasón Box, que ha construido la base de datos ODB y ha traducido de C++ a D2K el código de detección de inicios de eventos musicales.

También me gustaría expresar mi gratitud a los revisores y a todos los miembros del tribunal.

Este trabajo no habría sido posible sin la financiación del proyecto PROS-EMUS¹ y el programa de investigación Consolider Ingenio 2010 MIPRCV². También ha sido financiado por los proyectos españoles CICYT TAR³ y TIRIG⁴, y por los fondos FEDER de la Unión Europea y los proyectos de la Generalitat Valenciana GV04B-541 y GV06/166.

Dejando a un lado la investigación, me gustaría agradecer a mi familia y amigos (no voy a nombraros porque ya sabéis quiénes sois). A pesar de que no saben exactamente en qué trabajo y nunca se leerán un documento científico aburrido (y en inglés), su amistad y generosidad han sido cruciales durante este largo periodo.

Por último, esta tesis está dedicada a la persona más importante de mi vida, Teima, por su cariño y paciencia durante todo este tiempo.

1- Introducción⁶

La transcripción musical automática consiste en extraer las notas que están sonando (la partitura) a partir de una señal de audio digital. En el caso de la transcripción polifónica, se parte de señales de audio que pueden contener varias notas sonando simultáneamente.

Una partitura es una guía para interpretar información musical, y por tanto puede representarse de distintas maneras. La representación más extendida es la notación moderna usada en música tonal occidental. Para extraer una representación comprensible en dicha notación, además de las notas, sus tiempos de inicio y sus duraciones, es necesario indicar el tempo, la tonalidad y la métrica.

La aplicación más obvia de la extracción de la partitura es ayudar a un músico a escribir la notación musical a partir del sonido, lo cual es una tarea complicada cuando se hace a mano. Además de esta aplicación, la transcripción automática también es útil para otras tareas de recuperación de información musical, como detección de plagios, identificación de autor, clasificación de género, y asistencia a la composición cambiando la instrumentación o las notas para generar nuevas piezas musicales a partir de una ya existente. En general,

¹Código TIN2006-14932-C02

²Código CSD2007-00018

³Código TIC2000-1703-CO3-02

⁴Código TIC2003-08496-C04

⁶*Introduction.*

estos algoritmos también pueden proporcionar información sobre las notas para aplicar métodos que trabajan sobre música simbólica.

La transcripción musical automática es una tarea de recuperación de información musical en la que están implicadas varias disciplinas, tales como el procesamiento de señales, el aprendizaje automático, la informática, la psicoacústica, la percepción musical y la teoría musical.

Esta diversidad de factores provoca que haya muchas formas de abordar el problema. La mayoría de trabajos previos han utilizado diversos enfoques dentro del campo del procesamiento de la señal, aplicando metodologías para el análisis en el dominio de la frecuencia. En la literatura podemos encontrar múltiples algoritmos de separación de señales, sistemas que emplean algoritmos de aprendizaje y clasificación para detectar las notas, enfoques que consideran modelos psicoacústicos de la percepción del sonido, o sistemas que aplican modelos musicológicos como medida de coherencia de la detección.

La parte principal de un sistema de transcripción musical es el sistema de detección de frecuencias fundamentales, que determina el número de notas que están sonando en cada instante, sus alturas y sus tiempos de activación. Además del sistema de detección de frecuencias fundamentales, para obtener la transcripción completa de una pieza musical es necesario estimar el tempo a través de la detección de pulsos musicales, y obtener el tipo de compás y la tonalidad.

La transcripción polifónica es una tarea compleja que, hasta el momento, no ha sido resuelta de manera eficaz para todos los tipos de sonidos armónicos. Los mejores sistemas de detección de frecuencias fundamentales obtienen unos porcentajes de acierto del 60%, aproximadamente. Se trata, principalmente, de un problema de descomposición de señales en una mezcla, lo cual implica conocimientos avanzados sobre procesamiento de señales digitales, aunque debido a la naturaleza del problema también intervienen factores perceptuales, psicoacústicos y musicológicos.

El proceso de transcripción puede separarse en dos tareas: convertir una señal de audio en una representación de pianola, y convertir la pianola estimada en notación musical.

Muchos autores sólo consideran la transcripción automática como una conversión de audio a pianola, mientras que la conversión de pianola a notación musical se suele ver como un problema distinto. La principal razón de esto es que los procesos involucrados en la extracción de una pianola incluyen detección de alturas y segmentación temporal de las notas, lo cual es una tarea ya de por sí muy compleja. La conversión de pianola a partitura implica estimar el tempo, cuantizar el ritmo o detectar la tonalidad. Esta fase está más relacionada con la generación de una notación legible para los músicos.

A. RESUMEN

En general, un sistema de transcripción musical no es capaz de obtener con exactitud la partitura original que el músico ha interpretado. Las señales musicales normalmente son interpretaciones expresivas, más que traducciones literales de lo que puede leerse en una partitura. Por tanto, en una situación real, una partitura se puede interpretar de muchas formas distintas, y trasladar a una partitura las notas presentes en una señal de audio es un problema que no tiene una única solución.

Sin embargo, la conversión de una señal musical en una representación de pianola sin información rítmica ni armónica sólo depende de la información contenida en la forma de la onda. Más que una representación orientada a obtener partituras legibles, una pianola puede verse como una representación orientada a mostrar las frecuencias presentes en el sonido. La conversión de una señal en una pianola se puede hacer mediante un método de detección de frecuencias fundamentales. Este es el módulo principal de un sistema de transcripción polifónica, ya que estima el número de notas presentes en cada instante y sus alturas, inicios y duraciones. La mayoría de métodos para estimación de frecuencia fundamental en señales polifónicas tienen un alto coste computacional, debido a la complejidad de este problema.

La conversión de una pianola en una partitura legible requiere la extracción de información armónica y rítmica. La tonalidad está relacionada con la armonía, y las muestra relaciones de alturas en función de una tónica, que es el centro de gravedad armónico. La separación de instrumentos y su clasificación pueden usarse para identificar los distintos sonidos presentes en la señal, permitiendo la extracción de partituras individuales para cada instrumento. La estructura métrica se refiere a la organización temporal jerárquica, y especifica cuántos tiempos hay en cada compás y cuál es la duración de nota usada para representar un tiempo. De este modo, se pueden indicar los compases para obtener una partitura legible. El tempo es una medida para especificar la velocidad de ejecución de una pieza musical.

Cuando un músico interpreta una partitura, introduce desviaciones temporales, tanto intencionadas como involuntarias, y los inicios y duraciones de las notas deben ajustarse (cuantizarse) para obtener una representación legible. La presencia de estas desviaciones implica que, si se genera automáticamente un sonido a partir de una pianola, este no coincidirá exactamente con los tiempos de la partitura original. Esta es la razón principal por la que una pianola se considera una representación puramente orientada al contenido acústico.

Esta tesis está principalmente enfocada a resolver el problema de estimación de frecuencias fundamentales en señales polifónicas. Ésta es una tarea extremadamente complicada de resolver y que ha sido abordada en numerosas tesis doctorales.

Las principales contribuciones de este trabajo son un conjunto de nuevos métodos eficientes propuestos para la estimación de frecuencias fundamentales. Estos algoritmos se han evaluado y comparado con otros métodos, dando buenos resultados con un coste computacional muy bajo.

La detección de los comienzos de eventos musicales en señales de audio, o detección de onsets, también se ha abordado en este trabajo, desarrollando un método simple y eficiente para esta tarea. La información sobre onsets puede usarse para estimar el tempo o para refinar un sistema de detección de alturas.

Los métodos propuestos se han aplicado a otras tareas de recuperación de información musical, tales como clasificación de género, de modo, o identificación de la autoría de una obra. Para ello, se ha combinado características de audio con características simbólicas extraídas mediante la transcripción de señales musicales, usando métodos de aprendizaje automático para obtener el género, modo o autor.

2- Conocimientos previos⁸

Este capítulo introduce los conceptos necesarios para la adecuada comprensión del trabajo. Se describen los conceptos y términos relacionados con métodos de procesamiento de la señal, teoría musical y aprendizaje automático.

Primero, se hace una breve introducción de las distintas técnicas para el análisis de señales de audio basadas en la transformada de Fourier, incluyendo diferentes representaciones de tiempo-frecuencia.

A continuación, se analizan las propiedades de las señales musicales, y se clasifican los instrumentos con respecto a su mecanismo de generación del sonido y a sus características espectrales.

También se abordan los conceptos necesarios sobre teoría musical, describiendo las estructuras temporales y armónicas de la música occidental y su representación usando notación escrita y computacional.

Finalmente, se describen las técnicas basadas en aprendizaje automático que se han usado en este trabajo (redes neuronales y k vecinos más cercanos).

3 - Transcripción musical¹⁰

Este capítulo describe brevemente algunas características perceptuales relacionadas con el proceso que sigue un músico para realizar una transcripción musical. Seguidamente, se analizan las limitaciones teóricas de la transcripción automática desde un punto de vista del análisis y procesamiento de señales discretas.

⁸Background.

¹⁰Music transcription.

Finalmente, se introducen los conceptos básicos para la tarea de detección de onsets. También se describen y analizan las distintas métricas usadas para evaluar los sistemas de estimación de frecuencias fundamentales y detección de onsets.

4 - Estado de la cuestión¹²

En este capítulo se presenta una descripción de los distintos sistemas previos para estimación de una única frecuencia fundamental. Estos métodos se han clasificado en los que analizan la forma de onda en el dominio del tiempo, los que analizan la señal en el dominio de la frecuencia tras hacer una transformada de Fourier, los métodos basados en modelos de percepción acústica y los modelos probabilísticos.

Posteriormente, la revisión se extiende con una mayor cobertura a los métodos de estimación de varias frecuencias fundamentales simultáneas. Es complicado clasificar estos métodos usando una única taxonomía, ya que son muy complejos y por tanto suelen incluir diferentes técnicas de procesamiento. Por ejemplo, pueden categorizarse de acuerdo a su representación intermedia (dominio del tiempo, transformada de Fourier de tiempo corto, wavelets, bancos de filtros perceptuales, etc.), pero también respecto a su genericidad (algunos métodos necesitan información a-priori sobre el instrumento a transcribir, mientras que otros pueden usarse para analizar cualquier tipo de sonido armónico), a su capacidad para modelar distintos timbres (por ejemplo, los métodos paramétricos estadísticos pueden modelar envolventes cambiantes en el tiempo o la frecuencia como aquellas producidas por un saxo, mientras que los no paramétricos sólo pueden analizar patrones espectrales constantes, como los que produce un piano), o por el modo en que pueden abordar las interacciones entre distintos armónicos (métodos de estimación iterativa o conjunta).

En este trabajo, se ha propuesto una nueva categorización basada en la metodología principal que sigue el algoritmo, en lugar de la representación intermedia escogida en la taxonomía de estimación de una única frecuencia fundamental. Asimismo, también se han discutido y analizado los puntos fuertes y débiles para cada una de estas categorías.

Finalmente, se ha hecho lo propio con los sistemas de detección de onsets, clasificándolos en métodos de procesamiento de la señal y métodos de aprendizaje automático.

¹²*State of the art.*

5 - Detección de onsets usando un banco de filtros armónicos¹⁴

En este capítulo se propone un nuevo método para detección de onsets. La señal de audio se analiza usando un banco de filtros pasa-banda de un semitono, y se emplean las derivadas temporales de los valores filtrados para detectar variaciones espectrales relacionadas con el inicio de los eventos musicales.

Este método se basa en las características de los sonidos armónicos. Los primeros cinco armónicos de un sonido afinado coinciden con las frecuencias de otras notas en la afinación bien temperada usada en la música occidental. Otra característica importante de estos sonidos es que normalmente la mayor parte de su energía se concentra en los primeros armónicos.

El banco de filtros de un semitono está formado por un conjunto de filtros triangulares cuyas frecuencias centrales coinciden con las alturas musicales. En la fase de sostenimiento y relajación de una nota, puede haber ligeras variaciones en la intensidad y en la frecuencia de los armónicos. En este escenario, la comparación espectral directa puede generar falsos positivos.

En cambio, usando el banco de filtros propuesto, se minimizan los efectos de las variaciones espectrales sutiles que se producen durante las fases de sostenimiento y relajación de una nota, mientras que en el ataque se incrementan significativamente las amplitudes filtradas, ya que la mayor parte de energía de los parciales se concentra en las frecuencias centrales de estas bandas. De este modo, el sistema es especialmente sensible a variaciones de la frecuencia mayores de un semitono, y por tanto se tiene en cuenta las propiedades armónicas de los sonidos.

El método se ha evaluado y comparado con otros trabajos, dando buenos resultados dada su sencillez, y obteniendo una alta eficiencia. El algoritmo, desarrollado en C++, y la base de datos etiquetada para su evaluación se han hecho públicos para futuras investigaciones.

6 - Estimación de alturas usando métodos de aprendizaje supervisado¹⁶

En este capítulo se propone un método para la detección de alturas en piezas musicales interpretadas por un solo instrumento con un patrón espectral simple. Para ello, se parte de la hipótesis de que un paradigma de aprendizaje, tal como una red neuronal, es capaz de inferir un patrón espectral tras una fase de entrenamiento y, por tanto, detectar las notas en una pieza interpretada con el

¹⁴*Onset detection using a harmonic filter bank.*

¹⁶*Multiple pitch estimation using supervised learning methods.*

mismo instrumento con el que se ha entrenado el sistema. Se ha puesto a prueba dicha hipótesis, analizando su comportamiento y limitaciones con grabaciones sintéticas.

Por tanto, la hipótesis de trabajo es que un método supervisado puede ser capaz de aprender un patrón para un timbre determinado. Se ha entrenado el sistema usando la información de las alturas correctas y espectrogramas comprimidos mediante un banco de filtros de un semitono. Tras la fase de aprendizaje, el algoritmo supervisado es capaz de detectar los patrones de las alturas en el espectrograma, incluso en presencia de interferencias polifónicas (solapamiento armónico). Para evaluar este método, se han comparado los resultados usando redes neuronales dinámicas y k vecinos más cercanos.

El principal inconveniente de las técnicas de aprendizaje supervisado es que dependen de los datos de entrenamiento. La cantidad de variedad tímbrica y de combinaciones de distintas alturas pueden condicionar los resultados. Al ser complicado conseguir datos musicales reales alineados con sus correspondientes alturas, se han generado melodías sintetizadas en un escenario simplificado y restringido, obteniendo resultados prometedores para sonidos sintéticos afinados que poseen envolventes temporales constantes y patrones espectrales fijos.

Los resultados de la evaluación muestran que las redes neuronales son capaces de identificar y generalizar los patrones espectrales obteniendo muy buenos resultados para detección de alturas, mientras que los k vecinos no parecen adecuados para esta tarea, dado el inmenso espacio de posibles prototipos observables.

7 - Estimación de frecuencias fundamentales usando métodos de procesamiento de la señal¹⁸

Los métodos de estimación de frecuencias fundamentales basados en aprendizaje supervisado requieren datos de audio y simbólicos alineados para su entrenamiento. Por tanto, estos métodos dependen del conjunto de entrenamiento, y por este motivo muchos de ellos necesitan información a priori sobre el timbre a analizar. Probablemente, es posible que estos sistemas puedan llegar a generalizar e identificar correctamente las alturas en sonidos reales si se entrenan usando un conjunto de datos suficientemente amplio, pero aun así dependen de los datos de entrenamiento.

En grabaciones reales puede haber varios instrumentos sonando simultáneamente, que son desconocidos a priori, y que además suelen presentar patrones espectrales complejos. En este capítulo se describen tres métodos para detección de frecuencias fundamentales en señales polifónicas que están completamente

¹⁸ *Multiple fundamental frequency estimation using signal processing methods.*

basados en métodos de procesamiento de la señal, evitando así la necesidad de emplear un conjunto de entrenamiento.

El primero de estos métodos es un algoritmo de cancelación iterativa, principalmente enfocado a la transcripción de sonidos de cuerdas percutidas. El principal objetivo de este sistema es obtener una estimación básica de las frecuencias fundamentales presentes en señales reales, manteniendo un bajo coste computacional. Este método se ha integrado con éxito en un sistema más complejo para clasificación de géneros musicales.

Además del sistema de cancelación iterativa, se han propuesto dos nuevos métodos de estimación conjunta que son capaces de tener en cuenta las interacciones entre armónicos. Este tipo de métodos suele tener un alto coste computacional debido a la evaluación de muchas posibles combinaciones de alturas. Sin embargo, los métodos propuestos son muy eficientes. Estos métodos se han evaluado y comparado con otros trabajos en MIREX 2007 y MIREX 2008, obteniendo excelentes resultados con tiempos de ejecución muy bajos.

El primero de estos métodos realiza un análisis de la señal por ventanas, obteniendo un conjunto de alturas en cada instante. Para ello, primero se identifican los posibles candidatos a partir de los picos espectrales, y se generan todas las combinaciones de candidatos para que un algoritmo de estimación conjunta busque la mejor combinación teniendo en cuenta las interacciones entre armónicos.

Para evaluar una combinación, se construye una secuencia hipotética de parciales (HPS) para cada candidato. La puntuación de un candidato se calcula considerando la suma de las amplitudes de los armónicos de su HPS y una medida de suavidad de la envolvente espectral. La puntuación de una combinación se calcula como la suma al cuadrado de las puntuaciones de sus candidatos, y se selecciona la combinación de mayor puntuación en la ventana actual.

El método asume que las envolventes espectrales de los sonidos analizados tienden a variar suavemente en función de la frecuencia. El principio de suavidad espectral se ha usado anteriormente (aunque de distinta manera) en trabajos previos. La nueva medida de suavidad espectral está basada en la convolución de la secuencia hipotética de parciales con una ventana gaussiana.

Dada una combinación, la HPS de cada candidato se calcula teniendo en cuenta las interacciones entre los armónicos de todos los candidatos de la combinación. Para ello, primero se identifican los parciales solapados y se estiman sus amplitudes por interpolación lineal usando las amplitudes de los armónicos no solapados.

A diferencia del método de cancelación iterativa previamente descrito, que asume un patrón armónico constante, el método de estimación conjunta puede inferir patrones armónicos hipotéticos a partir de los datos espectrales,

evaluándolos de acuerdo a las propiedades de los sonidos armónicos. Esta metodología es adecuada para la mayoría de los sonidos armónicos, a diferencia del método de cancelación iterativa, el cual asume un patrón constante basado en los sonidos de instrumentos de cuerda percutida.

En este método de estimación conjunta, cada ventana se analiza de manera independiente, dando como resultado la combinación de frecuencias fundamentales que maximiza una puntuación. Una de sus principales limitaciones es que la información espectral que contiene una sola ventana se corresponde a un periodo temporal breve y, debido a la naturaleza de las señales musicales, en muchos casos es insuficiente para detectar las alturas de las notas, incluso para músicos expertos.

Partiendo de la hipótesis de que el contexto temporal es importante, se ha propuesto un segundo método de estimación conjunta, que extiende el anterior considerando información de ventanas adyacentes para producir una detección temporal suavizada. Adicionalmente, se ha incluido una técnica básica de seguimiento de frecuencias fundamentales usando para ello un grafo acíclico dirigido, para considerar de este modo más información contextual.

8 - Conclusiones y trabajo futuro²⁰

El objetivo principal de esta tesis es el desarrollo e implementación de un sistema de transcripción polifónica. La investigación se centra principalmente en la fase de detección de frecuencias fundamentales. Para lograr este objetivo, se han analizado otros sistemas previos y se han propuesto distintas alternativas basadas en el procesamiento de la señal de audio en el dominio de la frecuencia y en su descomposición y segmentación. Se intenta que los métodos propuestos, además de minimizar los errores, sean computacionalmente eficientes.

Para abordar el problema, se usan los datos de la señal en el dominio de la frecuencia haciendo un análisis por ventanas mediante la transformada de Fourier. Con estos datos de entrada se aplican distintas metodologías para extraer las notas que están presentes en la señal, usando algoritmos de aprendizaje (tales como vecinos más cercanos y redes neuronales dinámicas) y desarrollando nuevos métodos basados en el procesamiento de señales. Para ello, se tiene en cuenta la naturaleza armónica del sonido que genera la mayor parte de los instrumentos musicales (la estructura de sus patrones espectrales), que tratan de inferirse de la mezcla desarrollando técnicas de separación espectral y de reconocimiento de patrones.

Las principales contribuciones de este trabajo pueden resumirse en los siguientes puntos:

²⁰ *Conclusions and future work.*

- Una revisión exhaustiva del estado de la cuestión para detección de onsets y de frecuencias fundamentales en señales polifónicas. Los métodos existentes se han clasificado en funciones prominentes (*salience functions*), cancelación iterativa (*iterative cancellation*), estimación conjunta (*joint estimation*), aprendizaje supervisado (*supervised learning*), aprendizaje no supervisado (*unsupervised learning*), búsqueda de coincidencias (*matching pursuit*), modelos bayesianos (*Bayesian models*), modelos espectrales estadísticos (*statistical spectral models*), sistemas de pizarra (*blackboard systems*), y métodos de coincidencia con bases de datos (*database matching*). Se ha hecho un análisis de los puntos fuertes y débiles de cada una de estas categorías.
- El desarrollo de un sistema eficiente para la detección de onsets y la construcción de un conjunto de datos etiquetado para esta tarea. Las principales novedades en este campo son el uso de un banco de filtros de un doceavo de octava para comprimir la información armónica y las sencillas funciones de detección de onsets propuestas. El método presentado está principalmente indicado para la detección de onsets percusivos, ya que detecta cambios bruscos de energía, pero también considera las propiedades de las señales armónicas, lo cual hace que el sistema sea robusto ante variaciones espectrales producidas durante la fase de sostenimiento de los sonidos. El algoritmo ha sido evaluado y comparado con otros trabajos, obteniendo resultados satisfactorios.
- Dos nuevos métodos para la estimación de frecuencias fundamentales de instrumentos conocidos a priori usando técnicas de aprendizaje supervisado. Estos algoritmos fueron unos de los primeros métodos basados en aprendizaje automático que se han propuesto para esta tarea. Un banco de filtros armónicos se ha usado para reducir la cantidad de información espectral que se usa como entrada para una red neuronal de tiempo retardado preservando el principal contenido armónico. Se ha generado un conjunto de entrenamiento y validación para evaluar el método. Las conclusiones extraídas tras comparar para esta tarea los resultados obtenidos por los k vecinos más cercanos y las redes neuronales de tiempo retardado también son relevantes. La red neuronal claramente mejora los resultados obtenidos por los vecinos más cercanos usando sonidos sintetizados, mostrando las ventajas de las redes para la generalización en un espacio de observaciones muy extenso. Se han propuesto funciones de activación alternativas para generalizar los prototipos obtenidos mediante vecinos más cercanos, pero los resultados han seguido siendo claramente inferiores a los obtenidos con la red neuronal.

A. RESUMEN

- Un método sencillo de cancelación iterativa, principalmente orientado a la transcripción de música de piano con un coste computacional muy bajo. Se ha desarrollado un sistema completo para cargar un fichero de audio y que obtiene un fichero MIDI como resultado. Se ha propuesto una nueva arquitectura, basada en el análisis aislado de aquellos espectros que están tras cada onset detectado (para aumentar la eficiencia), y se ha evaluado la aplicación de un método para la extracción de sinusoides en esta tarea. El sistema propuesto ha sido la base para los métodos de estimación conjunta desarrollados con posterioridad, y ha sido usado con éxito para clasificación de género y otras tareas de recuperación de información musical.
- Métodos de estimación de frecuencias fundamentales en señales polifónicas basados en técnicas de procesamiento de señal para analizar música real sin ningún conocimiento a priori. Estos métodos, que son probablemente la mayor contribución de esta tesis, han alcanzado experimentalmente a los mejores algoritmos para esta tarea con un alto grado de eficiencia. Los patrones armónicos se han inferido de la información espectral, teniendo en cuenta la intensidad y una métrica propuesta para medir la suavidad de la envolvente en el dominio de la frecuencia, en un esquema de evaluación conjunta que tiene en cuenta las interacciones entre las distintas fuentes. Estos métodos también introducen un proceso de suavizado temporal considerando las combinaciones de alturas en ventanas adyacentes. También se ha propuesto un sistema sencillo de seguimiento de frecuencias fundamentales usando un grafo dirigido acíclico y ponderado. Estos nuevos métodos han conseguido tasas de acierto muy altas, aunque también poseen una alta eficiencia, que era la mayor barrera a la que se enfrentaban los métodos existentes de estimación conjunta.

Lineas de trabajo futuro

Como trabajo futuro, los métodos propuestos basados en redes neuronales podrían adaptarse fácilmente para detección de onsets. Al igual que en los sistemas de aprendizaje automático descritos para detección de frecuencias fundamentales, el banco de filtros de un doceavo de octava podría usarse para obtener los datos de entrada a la red, pero en el caso de esta tarea sólo una neurona sería necesaria en la capa de salida para clasificar cada ventana como onset o no onset. Este esquema de aprendizaje también implica ampliar la base de datos compilada para detección de onsets para obtener un conjunto de datos mayor y, por tanto, más fiable.

Los métodos de aprendizaje supervisado también podrían ser entrenados y evaluados usando mezclas de distintos instrumentos reales alineados con sus

correspondientes alturas etiquetadas. Como se ha descrito en este trabajo, no es una tarea sencilla alinear una base de datos de este tipo. Sin embargo, esta es una línea de investigación que debería ser explorada.

Probablemente, los sistemas de estimación de frecuencias fundamentales que sólo consideran ventanas individuales no podrán ser capaces de mejorar significativamente los resultados actuales. La cantidad de datos presente en el periodo correspondiente a una ventana no es suficiente para detectar las alturas, incluso para un músico experto. El contexto juega un papel muy importante en la música. Por ejemplo, es muy complicado identificar las alturas cuando escuchamos dos canciones que no están sincronizadas sonando simultáneamente, incluso si no son muy complejas.

Por tanto, la tarea de estimación de las alturas debería complementarse de algún modo con información temporal. La coherencia de las detecciones a lo largo del tiempo se ha considerado en uno de los métodos propuestos, pero podría extenderse usando un sistema fiable de seguimiento de frecuencias fundamentales. Sin embargo, esta tarea es complicada desde un punto de vista computacional.

La investigación futura de las arquitecturas alternativas propuestas para los métodos de estimación conjunta también es una línea de trabajo prometedora. La metodología de este sistema permite, por ejemplo, analizar conjuntamente combinaciones de aquellas ventanas que están entre dos onsets consecutivos, obteniendo las alturas para este intervalo. Perceptualmente, los resultados obtenidos con este esquema fueron mejores que analizando aisladamente conjuntos adyacentes de ventanas. Sin embargo, la menor resolución temporal y los errores en el método de detección de onsets, sumado al problema de detección de los offsets en el intervalo entre dos onsets, condicionan el porcentaje de acierto usando una métrica de evaluación clásica. Como trabajo futuro, está planeado evaluar estas arquitecturas usando una métrica perceptual.

La información multimodal también es una línea de trabajo prometedora. La inclusión de modelos musicales considerando la tonalidad, el tempo o la métrica para inferir probabilidades de notas podría complementar las estimaciones de las alturas. Estas líneas de trabajo están planeadas dentro del proyecto DRIMS en colaboración con el *Music Technology Group* de la Universitat Pompeu Fabra de Barcelona.

La transcripción musical interactiva también está planeada como trabajo futuro dentro del proyecto MIPRCV (Consolider Ingenio 2010). Se trata de desarrollar un método asistido por ordenador para transcripción musical. Usando un interfaz visual, los segmentos transcritos automáticamente pueden ser aceptados o corregidos por un músico experto. Después, estos segmentos validados pueden usarse como información para el método de estimación de

A. RESUMEN

frecuencias fundamentales, produciendo así sugerencias mejoradas y corrigiendo errores frecuentes.

La investigación futura de los métodos propuestos aplicados a otras tareas de recuperación de información musical, tales como clasificación de género o identificación de autor, también se está desarrollando en colaboración con el centro de investigación de *Software Technology and Interactive Systems (Vienna University of Technology)*.

Bibliography

- Abdallah, S. A. and Plumbley, M. D. (2003a). An ICA approach to automatic music transcription. In *Proc. 114th AES Convention*. (Cited on page 70).
- Abdallah, S. A. and Plumbley, M. D. (2003b). Probability as metadata: Event detection in music using ICA as a conditional density model. In *Proc. of the Fourth International Symposium on Independent Component Analysis (ICA)*, pages 233–238, Nara, Japan. (Cited on page 81).
- Abdallah, S. A. and Plumbley, M. D. (2004). Polyphonic music transcription by non-negative sparse coding of power spectra. In *Proc. of the 5th International Conference on Music Information Retrieval (ISMIR)*, pages 318–325, Barcelona, Spain. (Cited on pages 70 and 77).
- Ahmed, N., Natarjan, T., and Rao, K. (1974). Discrete cosine transform. *IEEE Trans. on Computers*, 23:90–93. (Cited on page 16).
- American Standards Association (1960). American standard acoustical terminology. Definition 12.9. Timbre. (Cited on page 19).
- Bay, M., Ehmann, A. F., and Downie, J. S. (2009). Evaluation of multiple-f0 estimation and tracking systems. In *Proc. of the 10th International Conference on Music Information Retrieval (ISMIR)*, pages 315–320. (Cited on pages xiii, 157, 162, 165, 166, and 167).
- Beauchamp, J. W., Maher, R. C., and Brown, R. (1993). Detection of musical pitch from recorded solo performances. In *Proc. 1993 Audio Engineering Society Convention*, pages 1–15, Berlin, Germany. Preprint 3541. (Cited on page 48).
- Bello, J. P. (2000). Blackboard system and top-down processing for the transcription of simple polyphonic music. In *Proc. of the COST G-6 Conference on Digital Audio Effects (DAFx)*, Verona, Italy. (Cited on pages 74, 75, and 77).
- Bello, J. P. (2004). *Towards the Automated Analysis of Simple Polyphonic Music: A Knowledge-based Approach*. PhD thesis, University of London, UK. (Cited on page 4).
- Bello, J. P., Daudet, L., Abdallah, S., Duxbury, C., Davies, M., and Sandler, M. B. (2005). A tutorial on onset detection in music signals. *IEEE Trans. on Speech and Audio Processing*, 13(5):1035–1047. (Cited on pages 52, 53, 77, and 78).

BIBLIOGRAPHY

- Bello, J. P., Daudet, L., and Sandler, M. (2002). Time-domain polyphonic transcription using self-generating databases. In *Proc. of the 112th Convention of the Audio Engineering Society*, Munich, Germany. (Cited on pages 75 and 123).
- Bello, J. P., Duxbury, C., Davies, M., and Sandler, M. (2004). On the use of phase and energy for musical onset detection in the complex domain. *IEEE Signal Processing Letters*, 11(6):553–556. (Cited on page 79).
- Bello, J. P. and Sandler, M. (2003). Phase-based note onset detection for music signals. In *International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, volume V, pages 441–444, Hong Kong. (Cited on page 79).
- Bertin, N., Badeau, R., and Richard, G. (2007). Blind signal decompositions for automatic transcription of polyphonic music: NMF and K-SVD on the benchmark. In *IEEE Int. Conf. on Acoustics, Speech and Signal Processing (ICASSP)*, volume I, pages 65–68, Honolulu, HI. (Cited on page 70).
- Bilmes, J. (1993). *Timing is of the Essence: Perceptual and Computational Techniques for Representing, Learning and Reproducing Expressive Timing in Percussive Rhythm*. MSc Thesis, MIT. (Cited on pages 32 and 85).
- Boulanger, R. (1999). *The CSound Book*. MIT Press, Cambridge, Massachusetts. (Cited on page 37).
- Bourlard, H. A. and Morgan, N. (1992). *Connectionist speech recognition. A Hybrid approach*. Kluwer Academic Publishers. (Cited on page 102).
- Bregman, A. S. (1990). *Auditory Scene Analysis*. MIT Press, Cambridge, MA. (Cited on page 44).
- Brossier, P. (2005). Fast onset detection using aubio. In [MIREX \(2005\)](#), onset detection contest. (Cited on page 92).
- Brossier, P., Bello, P., and Plumbley, D. (2004). Real-time temporal segmentation of note objects in music signals. In *Proc. of the International Computer Music Conference (ICMC)*, Florida. (Cited on pages 78 and 79).
- Brown, J. C. (1991). Calculation of a constant Q spectral transform. *Journal of the Acoustical Society of America*, 89(1):425–434. (Cited on page 14).
- Brown, J. C. (1992). Musical fundamental frequency tracking using a pattern recognition method. *Journal of the Acoustical Society of America*, 92(3):1394–1402. (Cited on page 58).

- Brown, J. C. and Puckette, M. S. (1992). An efficient algorithm for the calculation of a constant Q transform. *Journal of the Acoustical Society of America*, 92(5):2698–2701. (Cited on page 15).
- Burges, C. J. C. (1998). A tutorial on support vector machines for pattern recognition. In *Data Mining and Knowledge Discovery*, pages 121–167. Kluwer Academic Publishers, Boston. (Cited on page 69).
- Butterworth, S. (1930). On the Theory of Filter Amplifiers. *Wireless Engineer*, 7:536–541. (Cited on page 94).
- Cambouropoulos, E. (2000). From MIDI to traditional musical notation. In *Proc. of the AAAI 2000 Workshop on Artificial Intelligence and Music: Towards Formal Models for Composition, Performance and Analysis. 17th National Conference on Artificial Intelligence (AAAI)*, Austin, TX. (Cited on page 37).
- Cano, P. (1998). Fundamental frequency estimation in the SMS analysis. In *Proceedings of the Digital Audio Effects Workshop (DAFx)*. (Cited on page 59).
- Cao, C. and Li, M. (2008). Multiple F0 estimation in polyphonic music (MIREX 2008). In [MIREX \(2008\)](#), multiple f_0 estimation and tracking contest. (Cited on page 163).
- Cao, C., Li, M., Liu, J., and Yan, Y. (2007). Multiple f_0 estimation in polyphonic music. In [MIREX \(2007\)](#), multiple f_0 estimation and tracking contest. (Cited on pages 65 and 160).
- Carpenter, G. A., Grossberg, S., and Reynolds, J. (1991). ARTMAP: Supervised real-time learning and classification of nonstationary data by a self-organizing neural network. *Neural Networks*, 4:493–504. (Cited on page 102).
- Cañadas-Quesada, F. J., Vera-Candeas, P., Ruiz-Reyes, N., and Carabias-Orti, J. J. (2009). Polyphonic transcription based on temporal evolution of spectral similarity of gaussian mixture models. In *17th European Signal Processing Conference (EUSIPCO)*, pages 10–14, Glasgow, Scotland. (Cited on page 66).
- Cañadas-Quesada, F. J., Vera-Candeas, P., Ruiz-Reyes, N., Mata-Campos, R., and Carabias-Orti, J. J. (2008). Note-event detection in polyphonic musical signals based on harmonic matching pursuit and spectral smoothness. *Journal of New Music Research*, 37(3):167–183. (Cited on pages 71 and 127).
- Cemgil, A. T. (2004). *Bayesian Music Transcription*. PhD thesis, Radboud University of Nijmegen, Netherlands. (Cited on pages 4 and 72).

BIBLIOGRAPHY

- Cemgil, A. T., Kappen, B., and Barber, D. (2003). Generative model based polyphonic music transcription. In *IEEE Workshop on Applications of Signal Processing to Audio and Acoustics*, pages 181–184. (Cited on pages 1 and 72).
- Cemgil, A. T., Kappen, H. J., and Barber, D. (2006). A generative model for music transcription. *IEEE Trans. on Audio, Speech and Language Processing*, 14(2):679–694. (Cited on page 72).
- Chang, W. C., Su, A. W. Y., Yeh, C., Roebel, A., and Rodet, X. (2008). Multiple-F0 tracking based on a high-order HMM model. In *Proc. of the 11th Int. Conference on Digital Audio Effects (DAFx)*, Espoo, Finland. (Cited on pages 66, 168, and 173).
- Cohen, L. (1995). *Time-frequency analysis*. Prentice Hall. (Cited on page 66).
- Collins, N. (2005a). A change discrimination onset detector with peak scoring peak picker and time domain correction. In *MIREX (2005)*, onset detection contest. (Cited on page 99).
- Collins, N. (2005b). A comparison of sound onset detection algorithms with emphasis on psychoacoustically motivated detection functions. In *AES Convention 118*, pages 28–31, Barcelona. (Cited on pages 77 and 79).
- Comon, P. (1994). Independent component analysis, a new concept? *Signal processing*, 36:287–314. (Cited on page 70).
- Cont, A. (2006). Realtime multiple pitch observation using sparse non-negative constraints. In *Proc. of the 7th International Symposium on Music Information Retrieval (ISMIR)*, Victoria, Canada. (Cited on page 69).
- Cont, A. (2007). Real-time transcription of music signals: MIREX 2007 submission description. In *MIREX (2007)*, multiple f_0 estimation and tracking contest. (Cited on pages 159 and 160).
- Cont, A. (2008). *Modeling Musical Anticipation: From the time of music to the music of time*. PhD thesis, University of Paris VI and University of California in San Diego. (Cited on page 44).
- Cover, T. and Hart, P. (1967). Nearest neighbor pattern classification. *IEEE Trans. on Information Theory*, 13(1):21 – 27. (Cited on page 40).
- Daniel, A., Emiya, V., and David, B. (2008). Perceptually-based evaluation of the errors usually made when automatically transcribing music. In *Proc. of the 9th Int. Conference on Music Information Retrieval (ISMIR)*, pages 550–555, Philadelphia, PA. (Cited on page 52).

- Daubechies, I. (1988). Orthonormal bases of compactly supported wavelets. *Communications on Pure and Applied Mathematics*, 41(7):909–996. (Cited on page 13).
- Daubechies, I. (1992). *Ten Lectures on Wavelets*. Society for Industrial & Applied Mathematics. (Cited on page 13).
- Daudet, L. (2001). Transients modelling by pruned wavelet tress. In *Proc. of the Int. Computer Music Conference (ICMC)*, pages 18–21. (Cited on page 78).
- Davy, M. (2006a). An introduction to signal processing. In [Klapuri and Davy \(2006\)](#), chapter 2. (Cited on page 58).
- Davy, M. (2006b). Multiple fundamental frequency estimation based on generative models. In [Klapuri and Davy \(2006\)](#), chapter 7. (Cited on page 72).
- Davy, M. and Godsill, S. (2002). Detection of abrupt spectral changes using support vector machines. an application to audio signal segmentation. In *International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1313–1316. (Cited on page 81).
- Davy, M., Godsill, S. J., and Idier, J. (2006). Bayesian analysis of polyphonic western tonal music. *Journal of the Acoustical Society of America*, 119:2498–2517. (Cited on page 72).
- de Cheveigné, A. (1993). Separation of concurrent harmonic sounds: Fundamental frequency estimation and a time-domain cancellation model for auditory processing. *Journal of the Acoustical Society of America*, 93(6):3271–3290. (Cited on page 65).
- de Cheveigné, A. (2005). Pitch perception models. In Plack, C. J., Oxenham, A. J., Fay, R. R., and Popper, A. N., editors, *Pitch: neural coding and perception*, chapter 6. Springer. (Cited on pages 60 and 65).
- de Cheveigné, A. and Kawahara, H. (2002). YIN, a fundamental frequency estimator for speech and music. *Journal of the Acoustical Society of America*, 111(4):1917–1930. (Cited on pages 56 and 173).
- Deutsch, D. (1998). *The psychology of music, 2nd edition (cognition and perception)*. Academic press. (Cited on pages 26 and 198).
- Dijkstra, E. W. (1959). A note on two problems in connexion with graphs. *Numerische Mathematik*, pages 269–271. (Cited on page 139).

BIBLIOGRAPHY

- Dixon, S. (2006). Onset detection revisited. In *Proc. of the Int. Conf. on Digital Audio Effects (DAFx)*, pages 133–137, Montreal, Canada. (Cited on pages 77, 78, 79, 91, 92, and 140).
- Doval, B. (1994). *Estimation de la Fréquence Fondamentale des signaux sonores*. PhD thesis, Université Paris VI, Paris. (Cited on page 121).
- Doval, B. and Rodet, X. (1993). Fundamental frequency estimation and tracking using maximum likelihood harmonic matching and HMMs. In *International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, volume 1, pages 221–224. (Cited on page 58).
- Dubnowski, J. J., Schafer, R. W., and Rabiner, L. R. (1976). Real-time digital hardware pitch detector. *IEEE Trans. Acoustics, Speech, and Signal Processing (ASSP)*, 24:2–8. (Cited on page 55).
- Dubois, C. and Davy, M. (2005). Harmonic tracking using sequential Monte Carlo. In *IEEE/SP 13th Workshop on Statistical Signal Processing*, pages 1292–1296, Bordeaux, France. (Cited on page 72).
- Dubois, C. and Davy, M. (2007). Joint Detection and Tracking of Time-Varying Harmonic Components: A Flexible Bayesian Approach. *IEEE Trans. on Audio, Speech and Language Processing*, 15(4):1283–1295. (Cited on page 72).
- Duda, R., Lyon, R., and Slaney, M. (1990). Correlograms and the separation of sounds. In *Proc. IEEE Asilomar Conference on Signals, Systems and Computers*. (Cited on page 60).
- Duda, R. O., Hart, P. E., and Stork, D. G. (2000). *Pattern Classification*. John Wiley and Sons. (Cited on pages xi, 38, 39, 40, 41, and 104).
- Durrieu, J. L., Richard, G., and David, B. (2008). Singer melody extraction in polyphonic signals using source separation methods. In *Proc of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 169–172, Las Vegas, NV. (Cited on page 163).
- Duxbury, C., Sandler, M., and Davies, M. (2002). A hybrid approach to musical note onset detection. In *Proc. Digital Audio Effects Conference (DAFx)*, pages 33–38, Hamburg, Germany. (Cited on pages 78, 80, and 83).
- Egashira, K., Ono, N., and Sagayama, S. (2008). Sequential estimation of multiple fundamental frequencies through harmonic-temporal-structured clustering. In *MIREX (2008)*, multiple f_0 estimation and tracking contest. (Cited on pages 163 and 164).

- Ellis, D. P. W. (1996). *Prediction-driven computational auditory scene analysis*. PhD thesis, MIT. (Cited on page 74).
- Emiya, V. (2008). *Automatic transcription of piano music*. PhD thesis, Ecole Nationale Supérieure des Télécommunications (ENST), Paris, France. (Cited on page 4).
- Emiya, V., Badeau, R., and David, B. (2007). Multipitch estimation and tracking of inharmonic sounds in colored noise. In [MIREX \(2007\)](#), multiple f_0 estimation and tracking contest. (Cited on pages 159 and 160).
- Emiya, V., Badeau, R., and David, B. (2008a). Automatic transcription of piano music based on HMM tracking of jointly-estimated pitches. In [MIREX \(2008\)](#), multiple f_0 estimation and tracking contest. (Cited on pages 163 and 164).
- Emiya, V., Badeu, R., and David, B. (2008b). Automatic transcription of piano music based on HMM tracking of jointly-estimated pitches. In *Proc. European Signal Processing Conference (EUSIPCO)*, Rhodes, Greece. (Cited on pages 66, 123, 124, 128, 129, 159, and 160).
- Engelmore, R. S. and Morgan, A. J. (1988). *Blackboard Systems*. Addison-Wesley publishing. (Cited on page 74).
- Every, M. R. and Szymanski, J. E. (2006). Separation of synchronous pitched notes by spectral filtering of harmonics. *IEEE Trans. on Audio, Speech, and Language Processing*, 14(5):1845–1856. (Cited on pages 123 and 130).
- FitzGerald, D. (2004). *Automatic drum transcription and source separation*. PhD thesis, Dublin Inst. Technol. (Cited on pages 26 and 70).
- Fitzgerald, D. and Paulus, J. (2006). Unpitched percussion transcription. In [Klapuri and Davy \(2006\)](#), chapter 5. (Cited on page 26).
- Fletcher, H. and Munson, W. A. (1933). Loudness, its definition, measurement and calculation. *Journal of the Acoustical Society of America*, 5:82–108. (Cited on page 10).
- Fletcher, N. H. and Rossing, T. D. (1988). *The physics of musical instruments*. Springer, Berlin. (Cited on pages 23 and 124).
- Fonseca, N. and Ferreira, A. (2009). Measuring music transcription results based on a hybrid decay/sustain evaluation. In *Proc. of the 7th Triennial Conference of European Society for the Cognitive Sciences of Music (ESCOM)*, pages 119–124, Jyväskylä, Finland. (Cited on page 52).

BIBLIOGRAPHY

- Fraisse, P. (1998). Rhythm and tempo. In [Deutsch \(1998\)](#), chapter 6. (Cited on page [31](#)).
- Frigo, M. and Johnson, S. G. (2005). The design and implementation of FFTW3. *Proceedings of the IEEE*, 93(2):216–231. Special issue on “Program Generation, Optimization, and Platform Adaptation”. (Cited on pages [9](#) and [120](#)).
- Gabor, D. (1946). Theory of communication. *J. Inst. Elect. Eng.*, 93:429–457. (Cited on page [71](#)).
- Gabor, D. (1947). Acoustical quanta and the theory of hearing. *Nature*, 159(4044):591–594. (Cited on page [71](#)).
- Gerhard, D. (2003). Pitch extraction and fundamental frequency: History and current techniques. Technical report, University of Regina. (Cited on page [55](#)).
- Goto, M. (2000). A robust predominant f0 estimation method for real-time detection of melody and bass lines in cd recordings. In *Proc. of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, volume II, pages 757–760, Istanbul, Turkey. (Cited on pages [xii](#), [72](#), and [73](#)).
- Goto, M. (2003). RWC (Real World Computing) Music database. <http://staff.aist.go.jp/m.goto/RWC-MDB/>. (Cited on pages [10](#), [18](#), [21](#), [22](#), [25](#), [27](#), [38](#), [86](#), [88](#), [89](#), [90](#), and [157](#)).
- Goto, M. and Muraoka, Y. (1995). A real-time beat tracking system for audio signals. In *Proc. of International Computer Music Conference (ICMC)*, pages 171–174. (Cited on page [85](#)).
- Goto, M. and Muraoka, Y. (1996). Beat tracking based on multiple-agent architecture — a real-time beat tracking system for audio signals —. In *Proc. of the Second Int. Conf. on Multi-Agent Systems*, pages 103–110. (Cited on page [85](#)).
- Gouyon, F. (2008). *Computational rhythm description. A review and novel approach*. VDM Verlag Dr Müller. (Cited on page [31](#)).
- Grey, J. (1978). Timbre discrimination in musical patterns. *Journal of the Acoustical Society of America*, 64:467–472. (Cited on page [19](#)).
- Gribonval, R. and Bacry, E. (2003). Harmonic decomposition of audio signals with matching pursuit. *IEEE Trans. on Signal Processing*, 51(1):101–111. (Cited on page [71](#)).

- Griffin, D. W. and Lim, J. S. (1985). A new model-based speech analysis/synthesis system. In *Proc. of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, volume 10, pages 513–516, Tampa, FL. (Cited on page 121).
- Groble, M. (2008). Multiple fundamental frequency estimation. In *MIREX (2008)*, multiple f_0 estimation and tracking contest. (Cited on pages 75 and 163).
- Haar, A. (1911). Zur theorie der orthogonalen funktionensysteme. *Mathematische Annalen*, 71(1):38–53. (Cited on page 13).
- Hainsworth, S. (2003). *Techniques for the Automated Analysis of Musical Audio*. PhD thesis, Signal Processing Group, Department of Engineering, University of Cambridge. (Cited on pages xi, 4, 32, 44, and 76).
- Handel, S. (1989). *Listening: An introduction to the perception of auditory events*. Bradford Books / MIT Press. (Cited on page 31).
- Harris, F. J. (1978). On the use of windows for harmonic analysis with the discrete Fourier transform. *Proceedings of the IEEE*, 66(1):51–83. (Cited on page 11).
- Hartmann, W. M. (1996). Pitch, periodicity, and auditory organization. *Journal of the Acoustical Society of America*, 100(6):3491–3502. (Cited on page 21).
- Herrera, P., Klapuri, A., and Davy, M. (2006). Automatic classification of pitched musical instrument sounds. In *Klapuri and Davy (2006)*, chapter 6. (Cited on page 20).
- Hess, W. J. (1983). *Algorithms and Devices for Pitch Determination of Speech-Signals*. Springer-Verlag, Berlin. (Cited on page 55).
- Holland, J. H. (1992). *Adaptation in Natural and Artificial Systems: An Introductory Analysis with Applications to Biology, Control, and Artificial Intelligence*. MIT Press. (Cited on page 68).
- Honing, H. (2001). From time to time: The representation of timing and tempo. *Computer Music Journal*, 25(3):50–61. (Cited on page 31).
- Hornbostel, E. M. and Sachs, C. (1914). Systematik der musikinstrumente. *Zeitschrift für Ethnologie*, 46:553–590. (Cited on pages 20 and 24).
- Huang, X., Acero, A., and Hon, H. (2001). *Spoken Language Processing: A guide to theory, algorithm, and system development*. Prentice Hall. (Cited on pages 16 and 22).

BIBLIOGRAPHY

- Huron, D. (1989). Voice denumerability in polyphonic music of homogeneous timbres. *Music Perception*, 6(4):361–382. (Cited on page 51).
- Hush, D. R. and Horne, B. G. (1993). Progress in supervised neural networks. *IEEE Signal Processing Magazine*, 1(10):8–39. (Cited on page 39).
- Jensen, K. (1999). Envelope model of isolated musical sounds. In *Proc. Digital Audio Effects Conference (DAFx)*. (Cited on page 18).
- Juslin, P. N., Karlsson, J., Lindström, E., Friberg, A., and Schoonderwaldt, E. (2006). Play it again with feeling: Computer feedback in musical communication of emotions. *Journal of Experimental Psychology: Applied*, 12(2):79–95. (Cited on page 33).
- Kameoka, H., Nishimoto, T., and Sagayama, S. (2007). A multipitch analyser based on harmonic temporal structured clustering. *IEEE Trans. on Audio, Speech and Language Processing*, 5(3):982–994. (Cited on pages xii, 73, 74, 159, and 160).
- Kapanci, E. and Pfeffer, A. (2004). A hierarchical approach to onset detection. In *In Proc. International Computer Music Conference (ICMC)*, pages 438–441. (Cited on page 81).
- Kashino, K. and Godsill, S. J. (2004). Bayesian estimation of simultaneous musical notes based on frequency domain modeling. In *Int. Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, Montreal, Canada. (Cited on page 72).
- Kashino, K. and Tanaka, H. (1993). A sound source separation system with the ability of automatic tone modeling. In *Proc. of International Computer Music Conference (ICMC)*, pages 248–255. (Cited on page 72).
- Klapuri, A. (1998). Number theoretical means of resolving a mixture of several harmonic sounds. In *Proc. European Signal Processing Conference (EUSIPCO)*, Rhodes, Greece. (Cited on page 46).
- Klapuri, A. (1999). Sound onset detection by applying psychoacoustic knowledge. In *IEEE Int. Conf. on Acoustics, Speech and Signal Processing (ICASSP)*, pages 3089–3092, Phoenix, USA. (Cited on pages 77, 78, 79, and 83).
- Klapuri, A. (2001). Multipitch estimation and sound separation by the spectral smoothness principle. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 3381–3384, Salt Lake City, Utah. (Cited on page 63).

- Klapuri, A. (2003a). Multiple fundamental frequency estimation based on harmonicity and spectral smoothness. *IEEE Trans. on Speech and Audio Processing*, 11(6):804–816. (Cited on pages [xi](#), [47](#), [60](#), [63](#), [71](#), [120](#), [127](#), and [130](#)).
- Klapuri, A. (2003b). Musical meter estimation and music transcription. In *Proc. Cambridge Music Processing Colloquium*, pages 40–45. (Cited on pages [31](#) and [32](#)).
- Klapuri, A. (2004). *Signal processing methods for the automatic transcription of music*. PhD thesis, Tampere Univ. of Technology. (Cited on pages [4](#), [46](#), [62](#), [76](#), and [128](#)).
- Klapuri, A. (2005). A perceptually motivated multiple-f0 estimation method for polyphonic music analysis. In *IEEE workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA)*, New Paltz, NY. (Cited on page [64](#)).
- Klapuri, A. (2006a). Introduction to music transcription. In [Klapuri and Davy \(2006\)](#), chapter 1. (Cited on page [21](#)).
- Klapuri, A. (2006b). Multiple fundamental frequency estimation by summing harmonic amplitudes. In *Proc. of the Int. Conference on Music Information Retrieval (ISMIR)*, pages 216–221, Victoria, Canada. (Cited on pages [64](#), [120](#), [125](#), [141](#), and [149](#)).
- Klapuri, A. (2008). Multipitch analysis of polyphonic music and speech signals using an auditory model. *IEEE Trans. Audio, Speech and Language Processing*, 16(2):255–266. (Cited on page [64](#)).
- Klapuri, A. and Astola, J. (2002). Efficient calculation of a physiologically-motivated representation for sound. In *Proceedings of IEEE International Conference on Digital Signal Processing*. (Cited on page [60](#)).
- Klapuri, A. and Davy, M. (2006). *Signal processing methods for music transcription*. Springer. (Cited on pages [37](#), [195](#), [197](#), [199](#), [201](#), [209](#), and [212](#)).
- Klapuri, A., Eronen, A. J., and Astola, J. T. (2006). Analysis of the meter of acoustic musical signals. *IEEE Trans. on Audio, Speech and Language Processing*, 14(1):342–355. (Cited on page [32](#)).
- Klapuri, A., Virtanen, T., and Holm, J.-M. (2000). Robust multipitch estimation for the analysis and manipulation of polyphonic musical signals. In *Proc. COST-G6 Conference on Digital Audio Effects (DAFx)*, pages 233–236. (Cited on page [44](#)).

BIBLIOGRAPHY

- Kosuke, I., Ken'Ichi, M., and Tsutomu, N. (2003). Ear advantage and consonance of dichotic pitch intervals in absolute-pitch possessors. *Brain and cognition*, 53(3):464–471. (Cited on page 31).
- Krstulovic, S. and Gribonval, R. (2006). MPTK: Matching Pursuit made tractable. In *Proc. Int. Conf. on Acoustics, Speech, and Signal Processing (ICASSP)*, volume III, pages 496–499, Toulouse, France. (Cited on page 70).
- Krumhansl, C. (2004). The cognition of tonality - as we know it today. *Journal of New Music Research*, 33(3):253–268. (Cited on pages xi, 30, and 31).
- Lacoste, A. and Eck, D. (2005). Onset detection with artificial neural networks. In [MIREX \(2005\)](#), onset detection contest. (Cited on page 78).
- Lacoste, A. and Eck, D. (2007). A supervised classification algorithm for note onset detection. *EURASIP Journal on Advances in Signal Processing*. (Cited on page 81).
- Lahat, M., Niederjohn, R., and Krubsack, D. (1987). A spectral autocorrelation method for measurement of the fundamental frequency of noise-corrupted speech. *IEEE Trans. on Acoustics, Speech and Signal Processing*, 35(6):741 – 750. (Cited on page 57).
- Large, E. W. and Kolen, J. F. (1994). Resonance and the perception of musical meter. *Connection Science*, 6:279–312. (Cited on page 67).
- Lee, D. D. and Seung, H. (1999). Learning the parts of objects by non-negative matrix factorization. *Nature*, 401:788–791. (Cited on page 69).
- Lee, W.-C. and Kuo, C.-C. J. (2006). Musical onset detection based on adaptive linear prediction. *IEEE International Conference on Multimedia and Expo*, 0:957–960. (Cited on page 78).
- Lee, W.-C., Shiu, Y., and Kuo, C.-C. J. (2007). Musical onset detection with linear prediction and joint features. In [MIREX \(2007\)](#), onset detection contest. (Cited on page 79).
- Lerdahl, F. and Jackendoff, R. (1983). *A Generative Theory of Tonal Music*. MIT Press, Cambridge. (Cited on page 33).
- Leveau, P. (2007). A multipitch detection algorithm using a sparse decomposition with instrument-specific harmonic atoms. In [MIREX \(2007\)](#), multiple f_0 estimation and tracking contest. (Cited on page 160).
- Leveau, P., Vincent, E., Richard, G., and Daudet, L. (2008). Instrument-specific harmonic atoms for mid-level music representation. *IEEE Trans. on Audio, Speech, and Language Processing*, 16(1):116 – 128. (Cited on pages xi and 71).

- Li, Y. and Wang, D. L. (2007). Pitch detection in polyphonic music using instrument tone models. In *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, volume II, pages 481–484, Honolulu, HI. (Cited on page 74).
- Lidy, T., Grecu, A., Rauber, A., Pertusa, A., Ponce de León, P. J., and Iñesta, J. M. (2009). A multi-feature multi-classifier ensemble approach for audio music classification. In *MIREX (2009), audio genre classification contest*, Kobe, Japan. (Cited on page 4).
- Lidy, T., Rauber, A., Pertusa, A., and Iñesta, J. M. (2007). Improving genre classification by combination of audio and symbolic descriptors using a transcription system. In *Proc. of the 8th International Conference on Music Information Retrieval (ISMIR)*, pages 61–66, Vienna, Austria. (Cited on pages 4, 52, 119, 159, and 168).
- Lidy, T., Rauber, A., Pertusa, A., Ponce de León, P. J., and Iñesta, J. M. (2008). Audio Music Classification Using A Combination Of Spectral, Timbral, Rhythmic, Temporal And Symbolic Features. In *MIREX (2008), audio genre classification contest*, Philadelphia, PA. (Cited on page 4).
- Lloyd, L. S. (1970). *Music and Sound*. Ayer Publishing. (Cited on page 48).
- Maestre, E. and Gómez, E. (2005). Automatic characterization of dynamics and articulation of expressive monophonic recordings. In *AES Convention 118*. (Cited on page 18).
- Maher, R. C. (1989). *An Approach for the Separation of Voices in Composite Musical Signals*. PhD thesis, University of Illinois, IL, USA. (Cited on page 4).
- Maher, R. C. (1990). Evaluation of a method for separating digitized duet signals. *Journal of Audio Engineering Society*, 38:956–979. (Cited on pages 58, 65, and 130).
- Maher, R. C. and Beauchamp, J. W. (1994). Fundamental frequency estimation of musical signals using a two-way mismatch procedure. *Journal of the Acoustical Society of America*, 95:2254–2263. (Cited on pages xi, 58, and 59).
- Mallat, S. (1999). *A wavelet tour of signal processing*. Academic press, second edition. (Cited on page 13).
- Mallat, S. and Zhang, Z. (1993). Matching pursuits with time-frequency dictionaries. *IEEE Trans. on Signal Processing*, 41(12):3397–3415. (Cited on page 70).

BIBLIOGRAPHY

- Marolt, M. (2001). SONIC: transcription of polyphonic piano music with neural networks. In *Procs. of Workshop on Current Research Directions in Computer Music*. (Cited on page 101).
- Marolt, M. (2002). *Automatic Transcription of Piano Music with Neural Networks*. PhD thesis, University of Ljubljana, Slovenia. (Cited on page 4).
- Marolt, M. (2004a). A connectionist approach to automatic transcription of polyphonic piano music. *IEEE Trans. on Multimedia*, 6:439–449. (Cited on pages xi, 67, 69, 101, and 102).
- Marolt, M. (2004b). Networks of adaptive oscillators for partial tracking and transcription of music recordings. *Journal of New Music Research*, 33(1):49–59. (Cited on pages 67, 69, and 101).
- Marolt, M., Kavcic, A., and Privosnik, M. (2002). Neural networks for note onset detection in piano music. In *Proc. International Computer Music Conference (ICMC)*, Gothenburg, Sweden. (Cited on page 81).
- Martin, K. (1996). A blackboard system for automatic transcription of simple polyphonic music. Technical Report 385, MIT Media Lab. (Cited on pages xii, 74, 75, and 77).
- Masri, P. and Bateman, A. (1996). Improved modelling of attack transients in music analysis-resynthesis. In *Proc. of the International Computer Music Conference (ICMC)*, pages 100–103, Hong Kong. (Cited on page 24).
- Mcaulay, R. and Quatieri, T. (1986). Speech analysis/synthesis based on a sinusoidal representation. *IEEE Trans. on Acoustics, Speech, and Signal Processing*, 34(4):744–754. (Cited on page 23).
- McKay, C. (2003). Using blackboard systems for polyphonic transcription: A literature review. Course Paper, McGill University. (Cited on page 74).
- Meddis, R. and Hewitt, M. J. (1991a). Virtual Pitch and Phase Sensitivity of a Computer Model of the Auditory Periphery. I: Pitch Identification. *Journal of the Acoustical Society of America*, 89:2866–2882. (Cited on pages 59 and 60).
- Meddis, R. and Hewitt, M. J. (1991b). Virtual Pitch and Phase Sensitivity of a Computer Model of the Auditory Periphery. II: Phase sensitivity. *Journal of the Acoustical Society of America*, 89:2883–2894. (Cited on page 59).
- Meng, X. L. and Rubin, D. B. (1993). Maximum likelihood estimation via the ECM algorithm. *Biometrika*, 80(2):267–278. (Cited on page 74).

- Meredith, D. and Wiggins, G. A. (2005). Comparing pitch spelling algorithms. In *International Symposium of Music Information Retrieval (ISMIR)*, pages 280–287, London, UK. (Cited on page 36).
- Minsky, M. L. and Papert, S. (1969). *Perceptrons*. MIT Press, Cambridge, MA. (Cited on page 38).
- MIREX (2005). Music Information Retrieval Evaluation eXchange. Onset detection contest. <http://www.music-ir.org/evaluation/mirex-results/audio-onset/index.html>. (Cited on pages 93, 94, 192, 194, 202, and 208).
- MIREX (2006). Music Information Retrieval Evaluation eXchange. Onset detection contest. http://www.music-ir.org/mirex/2006/index.php/Audio_Onset_Detection_Results. (Cited on page 93).
- MIREX (2007). Music Information Retrieval Evaluation eXchange. Multiple fundamental frequency estimation and tracking contest. http://www.music-ir.org/mirex/2007/index.php/Multiple_Fundamental_Frequency_Estimation_%26_Tracking_Results. (Cited on pages xv, 52, 119, 127, 141, 157, 158, 159, 160, 161, 162, 167, 168, 193, 194, 197, 202, 207, and 212).
- MIREX (2007). Music Information Retrieval Evaluation eXchange. Onset detection contest. http://www.music-ir.org/mirex/2007/index.php/Audio_Onset_Detection_Results. (Cited on pages 93 and 202).
- MIREX (2008). Music Information Retrieval Evaluation eXchange. Multiple fundamental frequency estimation and tracking contest. http://www.music-ir.org/mirex/2008/index.php/Multiple_Fundamental_Frequency_Estimation_%26_Tracking_Results. (Cited on pages xv, 52, 119, 141, 157, 161, 162, 163, 164, 167, 168, 193, 196, 197, 199, 207, 208, 213, and 214).
- MIREX (2009). Music Information Retrieval Evaluation eXchange. Onset detection contest. http://www.music-ir.org/mirex/2009/index.php/Audio_Onset_Detection_Results. (Cited on pages 93, 94, 95, 98, 207, 209, and 211).
- Monti, G. and Sandler, M. B. (2002). Automatic polyphonic piano note extraction using fuzzy logic in a blackboard system. In *Proc. of the 5th Conference on Digital Audio Effects (DAFx)*, Hamburg, Germany. (Cited on page 74).
- Moon, T. K. (1996). The Expectation-Maximization Algorithm. *IEEE Signal Processing Magazine*, 13(6):47–60. (Cited on page 58).

BIBLIOGRAPHY

- Moore, B. C. J., editor (1995). *Hearing: Handbook of perception and cognition*. Academic Press, second edition. (Cited on page 17).
- Moore, B. C. J. (1997). *An introduction to the Psychology of Hearing*. Academic Press, fifth edition. (Cited on pages 10 and 53).
- Moorer, J. A. (1975). *On the Segmentation and Analysis of Continuous Musical Sound by Digital Computer*. PhD thesis, Dept. of Music, Stanford University. (Cited on pages 4 and 62).
- Moorer, J. A. (1977). Signal processing aspects of computer music: A survey. *Computer Music Journal*, 1(1):4–37. (Cited on page 23).
- Noll, M. (1967). Cepstrum pitch determination. *Journal of the Acoustical Society of America*, 41:293–309. (Cited on page 56).
- Noll, M. (1969). Pitch determination of human speech by the harmonic product spectrum, the harmonic sum spectrum, and a maximum likelihood estimate. In *Proc. of the Symposium on Computer Processing Communications*, pages 779–797, Polytechnic Institute of Brooklyn. (Cited on pages xi and 58).
- Noll, M. and Schroeder, M. R. (1964). Short-time “cepstrum” pitch detection. *Journal of the Acoustical Society of America*, 36:1030–1030. (Cited on page 56).
- Paiement, J. C., Grandvalet, Y., and Bengio, S. (2008). Predictive models for music. *Connection Science*, 21:253–272. (Cited on page 44).
- Patterson, R. D. (1982). Auditory filter shapes derived with noise stimuli. *Journal of the Acoustical Society of America*, 76:640–654. (Cited on page 17).
- Patterson, R. D., Allerhand, M., and Giguere, C. (1995). Time-domain modelling of peripheral auditory processing: A modular architecture and software platform. *Journal of the Acoustical Society of America*, 98:1890–1894. (Cited on page 17).
- Patterson, R. D., Nimmo-Smith, I., Weber, D. L., and Milroy, R. (1982). The deterioration of hearing with age: Frequency selectivity, the critical ratio, the audiogram, and speech threshold. *Journal of the Acoustical Society of America*, 72:1788–1803. (Cited on page 17).
- Peeters, G. (2004). A large set of audio features for sound description (similarity and classification) in the CUIDADO project. Technical report, IRCAM, Paris, France. (Cited on pages 18, 19, and 20).

- Peeters, G. (2006). Music pitch representation by periodicity measures based on combined temporal and spectral representations. In *Proc. of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, volume V, pages 53–56, Toulouse, France. (Cited on page 63).
- Pérez-Sancho, C. (2009). *Stochastic Language Models for Music Information Retrieval*. PhD thesis, Universidad de Alicante, Spain. (Cited on page 169).
- Pertusa, A. (2003). *Transcripción de melodías polifónicas mediante redes neuronales dinámicas*. MSc Thesis, Universidad de Alicante, Spain. (Cited on pages 101, 108, and 109).
- Pertusa, A. and Iñesta, J. M. (2004). Pattern recognition algorithms for polyphonic music transcription. In Fred, A., editor, *Pattern Recognition in Information Systems (PRIS)*, pages 80–89, Porto, Portugal. (Cited on page 101).
- Pertusa, A. and Iñesta, J. M. (2005). Polyphonic monotimbral music transcription using dynamic networks. *Pattern Recognition Letters*, 26(12):1809–1818. (Cited on page 101).
- Pertusa, A. and Iñesta, J. M. (2007). Multiple fundamental frequency estimation based on spectral pattern loudness and smoothness. In [MIREX \(2007\)](#), multiple f_0 estimation and tracking contest. (Cited on page 119).
- Pertusa, A. and Iñesta, J. M. (2008a). Multiple fundamental frequency estimation using Gaussian smoothness. In *Proc. of the IEEE Int. Conf. on Acoustics, Speech, and Signal Processing (ICASSP)*, pages 105–108, Las Vegas, NV. (Cited on pages 119, 159, and 160).
- Pertusa, A. and Iñesta, J. M. (2008b). Multiple fundamental frequency estimation using Gaussian smoothness and short context. In [MIREX \(2008\)](#), multiple f_0 estimation and tracking contest. (Cited on pages 119, 163, 164, and 167).
- Pertusa, A. and Iñesta, J. M. (2009). Note onset detection using one semitone filter-bank for MIREX 2009. In [MIREX \(2009\)](#), onset detection contest. (Cited on pages 84, 95, and 96).
- Pertusa, A., Klapuri, A., and Iñesta, J. M. (2005). Recognition of note onsets in digital music using semitone bands. *Lecture Notes in Computer Science*, 3773:869–879. (Cited on pages 84 and 92).
- Plumbley, M. D., Abdallah, S., Bello, J. P., Daview, M., Monti, G., and Sandler, M. (2002). Automatic music transcription and audio source separation. *Cybernetic and Systems*, 33(6):603–627. (Cited on pages 69, 70, and 74).

BIBLIOGRAPHY

- Poliner, G. E. and Ellis, D. P. W. (2007a). A discriminative model for polyphonic piano transcription. *EURASIP Journal on Advances in Signal Processing*. (Cited on pages [xi](#), [50](#), [51](#), [68](#), [157](#), [159](#), [160](#), [163](#), and [165](#)).
- Poliner, G. E. and Ellis, D. P. W. (2007b). Improving generalization for classification-based polyphonic piano transcription. In *Proc. of the IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA)*, pages 86–89, New Paltz, NY. (Cited on page [69](#)).
- Rabiner, L. (1989). A tutorial on hidden markov models and selected applications in speech recognition. *Proceedings of the IEEE*, 77(2):257–286. (Cited on page [61](#)).
- Rabiner, L., Cheng, M., Rosenberg, A., and McGonegal, C. (1976). A comparative performance study of several pitch detection algorithms. *IEEE Trans. on Acoustics, Speech and Signal Processing*, 24(5):399–418. (Cited on pages [55](#) and [57](#)).
- Raczynski, S. A., Ono, N., and Sagayama, S. (2007). Multipitch analysis with harmonic nonnegative matrix approximation. In *Proc. of the 8th Int. Conference on Music Information Retrieval (ISMIR)*, pages 381–386. (Cited on pages [70](#) and [160](#)).
- Raphael, C. (2002). Automatic transcription of piano music. In *Proc. Int. Symposium on Music Information Retrieval (ISMIR)*, pages 15–19. (Cited on page [72](#)).
- Reis, G., Fernandez, F., and Ferreira, A. (2008a). Genetic algorithm approach to polyphonic music transcription for MIREX 2008. In [MIREX \(2008\)](#), multiple f_0 estimation and tracking contest. (Cited on pages [163](#) and [164](#)).
- Reis, G., Fonseca, N., de Vega, F., and Ferreira, A. (2008b). A genetic algorithm based on gene fragment competition for polyphonic music transcription. *Lecture Notes in Computer Science*, 4974:305–314. (Cited on page [68](#)).
- Reis, G., Fonseca, N., Fernandez, F., and Ferreira, A. (2008c). A genetic algorithm approach with harmonic structure evolution for polyphonic music transcription. *IEEE International Symposium on Signal Processing and Information Technology (ISSPIT)*, pages 491 – 496. (Cited on pages [67](#) and [68](#)).
- Roads, C. (1996). *The Computer Music Tutorial*. MIT Press, Cambridge. (Cited on page [60](#)).
- Röbel, A. (2005). Onset detection in polyphonic signals by means of transient peak classification. In [MIREX \(2005\)](#), onset detection contest. (Cited on pages [80](#) and [94](#)).

- Röbel, A. (2009). Onset detection by means of transient peak classification in harmonic bands. In [MIREX \(2009\)](#), onset detection contest. (Cited on pages [94](#), [95](#), [96](#), and [97](#)).
- Rodet, X. (1997). Musical sound signals analysis/synthesis: Sinusoidal+residual and elementary waveform models. In *Proc. of the IEEE Time-Frequency and Time-Scale Workshop (TFTS'97)*, Coventry, GB. (Cited on page [121](#)).
- Rodet, X., Escribe, J., and Durignon, S. (2004). Improving score to audio alignment: Percussion alignment and precise onset estimation. In *Proc. of Int. Computer Music Conference (ICMC)*, pages 450–453. (Cited on page [53](#)).
- Ruiz-Reyes, N., Vera-Candeas, P., nadas Quesada, F. J. C., and Carabias, J. J. (2009). Fast communication: New algorithm based on spectral distance maximization to deal with the overlapping partial problem in note-event detection. *Signal Processing*, 89(8):1653–1660. (Cited on page [71](#)).
- Rumelhart, D. E., Hinton, G. E., and Williams, R. J. (1986). Learning representations by back-propagating errors. *Nature*, 323:533–536. (Cited on pages [39](#) and [104](#)).
- Ryynänen, M. (2006). Singing transcription. In [Klapuri and Davy \(2006\)](#), chapter 12. (Cited on page [26](#)).
- Ryynänen, M. (2008). *Automatic Transcription of Pitch Content in Music and Selected Applications*. PhD thesis, Tampere University of Technology. (Cited on page [4](#)).
- Ryynänen, M. and Klapuri, A. (2004). Modelling of note events for singing transcription. In *in Proc. ISCA Tutorial and Research Workshop on Statistical and Perceptual Audio*, page 6. MIT Press. (Cited on pages [xi](#), [56](#), [61](#), [64](#), and [65](#)).
- Ryynänen, M. and Klapuri, A. (2005). Polyphonic music transcription using note event modeling. In *Proc. IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA)*, pages 319–322, New Paltz, New York, USA. (Cited on pages [xi](#), [51](#), [64](#), [74](#), [120](#), [159](#), [160](#), [163](#), [164](#), [167](#), and [169](#)).
- Sachs, C. (1940). *The history of Musical Instruments*. Norton, New York. (Cited on page [20](#)).
- Sano, H. and Jenkins, B. K. (1989). A neural network model for pitch perception. *Computer Music Journal*, 13(3):41–48. (Cited on page [102](#)).

BIBLIOGRAPHY

- Scheirer, E. D. (1998). Tempo and beat analysis of acoustic musical signals. *Journal of the Acoustical Society of America*, 103(1):588–601. (Cited on pages 83 and 85).
- Schmidt, M. N. (2008). *Single-channel source separation using non-negative matrix factorization*. PhD thesis, Technical University of Denmark. (Cited on page 70).
- Schouten, J. F. (1940). The residue and the mechanism of hearing. In *Proceedings Koninklijke Nederlandse Akademie van Wetenschappen*, volume 43, pages 991–999. (Cited on page 22).
- Schwefel, H. P. (1995). *Evolution and Optimum Seeking*. Wiley & Sons, New York. (Cited on page 66).
- Selfbridge-Field, E. (1997). *Beyond MIDI: the handbook of musical codes*. MIT Press, Cambridge, USA. (Cited on page 37).
- Serra, X. (1997). Musical sound modeling with sinusoids plus noise. In Roads, C., Pope, S. T., Picialli, A., and De Poli, G., editors, *Musical signal processing*, pages 91–122. Swets and Zeitlinger. (Cited on pages 23 and 59).
- Shannon, B. J. and Paliwal, K. K. (2003). A comparative study of filter bank spacing for speech recognition. In *Proc. Microelectronic Engineering Research Conference*. (Cited on page 16).
- Slaney, M. (1993). An efficient implementation of the Patterson-Holdsworth auditory filter bank. Technical Report 35, Perception Group, Advanced Technology Group, Apple Computer, Inc. (Cited on page 17).
- Sloboda, J. A. (1985). *The musical mind. The cognitive psychology of music*. Oxford: The Clarendon Press. (Cited on page 43).
- Smaragdis, P. (2001). *Redundancy Reduction for Computational Audition, a Unifying Approach*. PhD thesis, MAS Department, MIT. (Cited on page 44).
- Smaragdis, P. and Brown, J. (2003). Non-negative matrix factorization for polyphonic music transcription. In *Proc. IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA)*, New Paltz, NY. (Cited on pages xi, 69, and 70).
- Sondhi, M. M. (1968). New methods of pitch extraction. *IEEE Trans. Audio Electroacoust.*, 16:262–266. (Cited on page 56).

- Sterian, A. D. (1999). *Model-Based Segmentation of Time-Frequency Images for Musical Transcription*. PhD thesis, University of Michigan. (Cited on page 72).
- Stevens, S., Volkman, J., and Newman, E. (1937). A scale for the measurement of the psychological magnitude of pitch. *Journal of the Acoustical Society of America*, 8(3):185–190. (Cited on page 16).
- Stowell, D. and Plumbley, M. D. (2007). Adaptive whitening for improved real-time audio onset detection. In *Proc. of the Int. Computer Music Conference (ICMC)*, pages 312–319. (Cited on pages 78 and 79).
- Sundberg, J. (1987). *The science of singing voice*. Northern Illinois University Press. (Cited on page 26).
- Tan, H. L., Zhu, Y., and Chaisorn, L. (2009). An energy-based and pitch-based approach to audio onset detection. In [MIREX \(2009\)](#), onset detection contest. (Cited on pages 52, 82, 94, 95, 96, 97, and 99).
- Taylor, I. and Greenhough, M. (1993). An object oriented ARTMAP system for classifying pitch. In *Proceedings of the International Computer Music Conference (ICMC)*, pages 244–247, Tokyo, Japan. (Cited on page 102).
- Tolonen, A. and Karjalainen, M. (2000). A computationally efficient multipitch analysis model. *IEEE Trans. on Speech and Audio Processing*, 8(6):708–716. (Cited on pages 60 and 62).
- Tzanetakis, G. (2009). Marsyas submissions to MIREX 2009. In [MIREX \(2009\)](#), onset detection contest. (Cited on pages 93, 95, and 96).
- Tzanetakis, G., Essl, G., and Cook, P. (2001). Audio analysis using the discrete wavelet transform. In *Proc. Conf. in Acoustics and Music Theory Applications (WSES)*. (Cited on page 13).
- Vercoe, B. (1991). *The CSound Reference Manual*. MIT Press, Cambridge, Massachusetts. (Cited on page 37).
- Verma, T. S. and Meng, T. H. Y. (2000). Extending spectral modeling synthesis with transient modeling synthesis. *Computer Music Journal*, 24(2):47–59. (Cited on page 24).
- Vidal, E., Casacuberta, F., Rodríguez, L., Civera, J., and Martínez, C. D. (2006). Computer-assisted translation using speech recognition. *IEEE Trans. on Audio, Speech and Language Processing*, 14(3):941–951. (Cited on page 173).

BIBLIOGRAPHY

- Viitaniemi, T., Klapuri, A., and Eronen, A. (2003). A probabilistic model for the transcription of single-voice melodies. In *Proc. of the Finnish Signal Processing Symposium (FINSIG)*, pages 59–63. (Cited on page 61).
- Vincent, E. (2004). *Modèles d'instruments pour la séparation de sources et la transcription d'enregistrements musicaux*. PhD thesis, Université Paris VI. (Cited on page 4).
- Vincent, E., Bertin, N., and Badeau, R. (2007). Two nonnegative matrix factorization methods for polyphonic pitch transcription. In *MIREX (2007)*, multiple f_0 estimation and tracking contest. (Cited on pages 70, 159, 160, 163, and 164).
- Vincent, E. and Plumbley, M. D. (2005). A prototype system for object coding of musical audio. In *Proc. IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA)*, pages 249–242, New Paltz, NY. (Cited on page 72).
- Vincent, E. and Rodet, X. (2004). Music transcription with ISA and HMM. In *Proc. 5th International Conference on Independent Component Analysis and Blind Signal Separation*, pages 1197–1204. (Cited on page 72).
- Virtanen, T. (2000). *Audio signal modeling with sinusoids plus noise*. MSc Thesis, Tampere University of Technology. (Cited on pages 121 and 122).
- Virtanen, T. (2006). Unsupervised learning methods for source separation. In *Klapuri and Davy (2006)*, chapter 9. (Cited on page 70).
- Virtanen, T. (2007). Monaural sound source separation by nonnegative matrix factorization with temporal continuity and sparseness criteria. *IEEE Trans. on Audio, Speech and Language Processing*, 15(3):1066–1074. (Cited on page 70).
- Viterbi, A. (1967). Error bounds for convolutional codes and an asymptotically optimum decoding. *IEEE Trans. on Information Theory*, 13(2):260–269. (Cited on page 61).
- Vos, P. and Vianen, B. G. V. (1984). Thresholds for discrimination between pure and tempered intervals: The relevance of nearly coinciding harmonics. *Journal of the Acoustical Society of America*, 77:176–187. (Cited on page 30).
- Waibel, A. (1989). Modular construction of time-delay neural networks for speech recognition. *Neural Computation*, 1:39–46. (Cited on pages 39 and 102).

- Walmsley, P., Godsill, S., and Rayner, P. (1999). Polyphonic pitch tracking using joint bayesian estimation of multiple frame parameters. In *Proc. IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA)*, pages 119–122, New Paltz, NY. (Cited on page 72).
- Wan, J., Wu, Y., and Dai, H. (2005). A harmonic enhancement based multipitch estimation algorithm. In *IEEE International Symposium on Communications and Information Technology (ISCIT) 2005*, volume 1, pages 772 – 776. (Cited on page 65).
- Wang, W., Luo, Y., Chambers, J. A., and Sanei, S. (2008). Note onset detection via nonnegative factorization of magnitude spectrum. *EURASIP Journal on Advances in Signal Processing*. (Cited on page 81).
- Wessel, D. L. (1979). Timbre space as a musical control structure. *Computer Music Journal*, 3(2):45–52. (Cited on page 19).
- Wood, A. (2008). *The physics of music*. Davies Press. (Cited on page 47).
- Woodruff, J., Li, Y., and Wang, D. (2008). Resolving overlapping harmonics for monoaural musical sound separation using pitch and common amplitude modulation. In *Proc. of the International Symposium on Music Information Retrieval (ISMIR)*, pages 538–543, Philadelphia, PA. (Cited on page 130).
- Yeh, C. (2008). *Multiple fundamental frequency estimation of polyphonic recordings*. PhD thesis, Universite Paris VI - Pierre et Marie Curie. (Cited on pages xi, 4, 45, 46, 48, 65, 76, 128, 160, and 173).
- Yeh, C., Röbel, A., and Rodet, X. (2005). Multiple fundamental frequency estimation of polyphonic music signals. In *IEEE, Int. Conf. on Acoustics, Speech and Signal Processing (ICASSP)*, volume III, pages 225–228, Philadelphia, PA. (Cited on pages 65, 126, 127, 129, and 130).
- Yeh, C. and Roebel, A. (2006). Adaptive noise level estimation. In *Proc. of the 9th Int. Conference on Digital Audio Effects (DAFx)*, Montreal, Canada. (Cited on page 65).
- Yeh, C. and Roebel, A. (2009). The expected amplitude of overlapping partials of harmonic sounds. In *Proc. of the International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Taipei, Taiwan. (Cited on page 47).
- Yeh, C., Roebel, A., and Chang, W. C. (2008). Multiple F0 estimation for MIREX 08. In *MIREX (2008)*, multiple f_0 estimation and tracking contest. (Cited on pages 163, 164, and 167).

BIBLIOGRAPHY

- Yeh, C., Roebel, A., and Rodet, X. (2006). Multiple f_0 tracking in solo recordings of monodic instruments. In *Proc. of the 120th AES Convention*, Paris, France. (Cited on pages 48 and 66).
- Yin, J., Sim, T., Wang, Y., and Shenoy, A. (2005). Music transcription using an instrument model. In *Proc. of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, volume III, pages 217–220. (Cited on page 65).
- Young, S., Kershaw, D., Odell, J., Ollason, D., Valtchev, V., and Woodland, P. (2000). *The HTK book (for HTK version 3.1)*. Cambridge University. (Cited on page 87).
- Young, S. J., Russell, N. H., and Thornton, J. H. S. (1989). Token passing: a simple conceptual model for connected speech recognition systems. Technical report, Cambridge University Engineering Department. (Cited on page 65).
- Zhou, R. (2006). *Feature extraction of Musical Content For Automatic Music Transcription*. PhD thesis, École Polytechnique Fédérale de Lausanne. (Cited on pages 4 and 69).
- Zhou, R. and Mattavelli, M. (2007). A new time-frequency representation for music signal analysis: Resonator time-frequency image. In *Proc. Int. Conference on Information Sciences, Signal Processing and its Applications*, Sharjah, U. Arab Emirates. (Cited on pages 63 and 80).
- Zhou, R., Mattavelli, M., and Zoia, G. (2008). Music Onset Detection Based on Resonator Time Frequency Image. *IEEE Transactions On Audio, Speech And Language Processing*, 16(8):1685–1695. (Cited on page 80).
- Zhou, R. and Reiss, J. D. (2008). A real-time polyphonic music transcription system. In *MIREX (2008)*, multiple f_0 estimation and tracking contest. (Cited on page 164).
- Zhou, R., Reiss, J. D., Mattavelli, M., and Zoia, G. (2009). A computationally efficient method for polyphonic pitch estimation. *EURASIP Journal on Advances in Signal Processing*, (28). (Cited on pages 63, 127, and 160).
- Zhu, Y. and Kankanhalli, M. (2006). Precise pitch profile feature extraction from musical audio for key detection. *IEEE Trans. on Multimedia*, 8(3):575–584. (Cited on pages 94 and 99).
- Zwicker, E. (1961). Subdivision of the audible frequency range into critical bands (Frequenzgruppen). *Journal of the Acoustical Society of America*, 33(2):248. (Cited on page 16).

BIBLIOGRAPHY

- Zwicker, E., Flottorp, G., and Stevens, S. S. (1957). Critical bandwidth in loudness summation. *Journal of the Acoustical Society of America*, 29:548–557. (Cited on page 16).

