Tampereen teknillinen yliopisto. Julkaisu 626
Tampere University of Technology. Publication 626

Tuomas Virtanen

Sound Source Separation in Monaural Music Signals

Thesis for the degree of Doctor of Technology to be presented with due permission for public examination and criticism in Tietotalo Building, Auditorium TB109, at Tampere University of Technology, on the 3rd of November 2006, at 12 noon.

Tampereen teknillinen yliopisto - Tampere University of Technology
Tampere 2006

ISBN 952-15-1667-4
ISSN 1459-2045

Abstract

Sound source separation refers to the task of estimating the signals produced by individual sound sources from a complex acoustic mixture. It has several applications, since monophonic signals can be processed more efficiently and flexibly than polyphonic mixtures.

This thesis deals with the separation of monaural, or, one-channel music recordings. We concentrate on separation methods, where the sources to be separated are not known beforehand. Instead, the separation is enabled by utilizing the common properties of real-world sound sources, which are their continuity, sparseness, and repetition in time and frequency, and their harmonic spectral structures. One of the separation approaches taken here use unsupervised learning and the other uses model-based inference based on sinusoidal modeling.

Most of the existing unsupervised separation algorithms are based on a linear instantaneous signal model, where each frame of the input mixture signal is modeled as a weighted sum of basis functions. We review the existing algorithms which use independent component analysis, sparse coding, and non-negative matrix factorization to estimate the basis functions from an input mixture signal.

Our proposed unsupervised separation algorithm based on the instantaneous model combines non-negative matrix factorization with sparseness and temporal continuity objectives. The algorithm is based on minimizing the reconstruction error between the magnitude spectrogram of the observed signal and the model, while restricting the basis functions and their gains to non-negative values, and the gains to be sparse and continuous in time. In the minimization, we consider iterative algorithms which are initialized with random values and updated so that the value of the total objective cost function decreases at each iteration. Both multiplicative update rules and a steepest descent algorithm are proposed for this task. To improve the convergence of the projected steepest descent algorithm, we propose an augmented divergence to measure the reconstruction error. Simulation experiments on generated mixtures of pitched instruments and drums were run to monitor the behavior of the proposed method. The proposed method enables average signal-to-distortion ratio (SDR) of 7.3 dB, which is higher than the SDRs obtained with the other tested methods based on the instantaneous signal model.

To enable separating entities which correspond better to real-world sound objects, we propose two convolutive signal models which can be used to represent

time-varying spectra and fundamental frequencies. We propose unsupervised learning algorithms extended from non-negative matrix factorization for estimating the model parameters from a mixture signal. The objective in them is to minimize the reconstruction error between the magnitude spectrogram of the observed signal and the model while restricting the parameters to non-negative values. Simulation experiments show that time-varying spectra enable better separation quality of drum sounds, and time-varying frequencies representing different fundamental frequency values of pitched instruments conveniently.

Another class of studied separation algorithms is based on the sinusoidal model, where the periodic components of a signal are represented as the sum of sinusoids with time-varying frequencies, amplitudes, and phases. The model provides a good representation for pitched instrument sounds, and the robustness of the parameter estimation is here increased by restricting the sinusoids of each source to harmonic frequency relationships.

Our proposed separation algorithm based on sinusoidal modeling minimizes the reconstruction error between the observed signal and the model. Since the rough shape of spectrum of natural sounds is continuous as a function of frequency, the amplitudes of overlapping overtones can be approximated by interpolating from adjacent overtones, for which we propose several methods. Simulation experiments on generated mixtures of pitched musical instruments show that the proposed methods allow average SDR above 15 dB for two simultaneous sources, and the quality decreases gradually as the number of sources increases.

Preface

This work has been carried out at the Institute of Signal Processing, Tampere University of Technology, during 2001-2006. I wish to express my deepest gratitude to my supervisor Professor Anssi Klapuri for guiding and encouraging me in my research work. I would like to thank the past and present members of the Audio Research Group, especially Jouni Paulus, Matti Ryynänen, and Antti Eronen, for creating an inspirational and relaxed atmosphere for our work. I also would like to thank my former supervisor Professor Jaakko Astola and other staff of the Institute of Signal Processing, whose work has resulted in an excellent environment for research.

Several people have reviewed parts of this thesis during the writing process, and their comments have helped me improve the thesis. In addition to my colleagues, I wish to thank Manuel Davy, Derry FitzGerald, Michael Casey, the peer reviewers of my journal manuscripts, and the preliminary assessors of the thesis, Prof. Vesa Välimäki and Prof. Dan Ellis.

The financial support provided by Tampere Graduate School in Information Science and Engineering, Nokia Foundation, and Academy of Finland is gratefully acknowledged.

I wish to thank my parents for their support in all my efforts. My warmest thanks go to my love Virpi, who has helped me carry on the sometimes hard process of completing this thesis.

Contents

List of Acronyms

AAC	Advanced Audio Coding
DFT	discrete Fourier transform
ICA	independent component analysis
IDFT	inverse discrete Fourier transform
ISA	independent subspace analysis
HMM	hidden Markov model
LPC	linear prediction coding
MAP	maximum a posteriori
MFCC	Mel-frequency cepstral coefficient
MFFE	multiple fundamental frequency estimator
MIDI	Musical Instrument Digital Interface
MLE	maximum likelihood estimator
MPEG	Moving Picture Experts Group
NLS	non-negative least squares
NMD	non-negative matrix deconvolution
NMF	non-negative matrix factorization
PCA	principal component analysis
SDR	signal-to-distortion ratio
STFT	short-time Fourier transform
SVD	singular value decomposition
SVM	support vector machine

Chapter 1

Introduction

Computational analysis of audio signals where multiple sources are present is a challenging problem. The acoustic signals of the sources mix, and the estimation of an individual source is disturbed by other co-occurring sounds. This could be solved by using methods which separate the signals of individual sources from each other, which is defined as sound source separation.

There are many signal processing tasks where sound source separation could be utilized, but the performance of the existing algorithms is quite limited compared to the human auditory system, for example. Human listeners are able to perceive individual sources in complex mixtures with ease, and several separation algorithms have been proposed that are based on modeling the source segregation ability in humans.

A recording done with multiple microphones enables techniques which use the spatial location of the source in the separation [46, 177], which often makes the separation task easier. However, often only a single-channel recording is available, and for example music is usually distributed in stereo (two-channel) format, which is not sufficient for spatial location based separation except only in few trivial cases.

This thesis deals with the source separation of one-channel music signals. We concentrate on separation methods which do not use source-specific prior knowledge, i.e., they are not trained for a specific source. Instead, we try to find general properties of music signals which enable the separation. Two different separation strategies are studied. First, the temporal continuity and redundancy of the musical sources is used to design *unsupervised learning* algorithms to learn the sources from a long segment of an audio signal. Second, the harmonic spectral structure of pitched musical instruments is used to design a parametric model based on a sinusoidal representation, which enables *model-based inference* in the estimation of the sources in short audio segments.

1.1 Problem Definition

When several sound sources are present simultaneously, the acoustic waveform $x(n)$ of the observed time-domain signal is the superposition of the source signals $s_m(n)$:

$$x(n) = \sum_{m=1}^{M} s_m(n), \qquad n = 1, \ldots, N \tag{1.1}$$

where s_m is the m^{th} source signal at time n, and M is the number of sources.

Sound source separation is defined as the task of recovering one or more source signals $s_m(n)$ from $x(n)$. Some algorithms concentrate on separating only a single source, whereas some try to separate them all. The term *segregation* has also been used as a synonym for separation.

A complication with the above definition is that there does not exist a unique definition for a sound source. One possibility is to consider each vibrating physical entity, for example each musical instrument, as a sound source. Another option is to define this according to what humans tend to perceive as a single source: for example, if a violin section plays in unison,[1] the violins are perceived as a single source, and usually there is no need to separate their signals from each other. In [93, pp. 302-303], these two alternatives are referred to as physical sound and perceptual sound, respectively. Here we do not specifically commit ourselves to either of these. Usually the type of the separated sources is determined by the properties of the algorithm used, and this can be partly affected by the designer according to the application at hand.

In some separation applications, prior information about the sources may be available. For example, the source instruments can be defined manually by the user, and in this case it is usually advantageous to optimize the algorithm by using training signals where each instrument is present in isolation.

In general, source separation without prior knowledge of the sources is referred to as *blind source separation* [77, pp. 2-4], [86, pp. 3-6]. Since the work in this thesis uses prior knowledge of the properties of music signals, the term blind is not used here. In the most difficult blind source separation problems even the number of sources is unknown.

Since the objective in the separation is to estimate several source signals from one input signal, the problem is underdetermined if there are no assumptions made about the signals. When there is no source-specific prior knowledge, the assumption can be for example that the sources are statistically independent, or that certain source statistics (for example the power spectra) do not change over time. It is not not known what are the sufficient requirements and conditions to enable their estimation; it is likely that in a general case we cannot separate the exact time-domain signals, but only achieve extraction of some source features. Part of this process is calculating a representation of the signal, where the important features for the application at hand are retained, and unnecessary information is discarded. For example, many of the unsupervised algorithms

[1] Unison is a musical term which means that instruments play the same pitch.

discussed in Chapter 2 operate on magnitude spectrograms, and are not able to estimate phases of the signals. The human audio perception can also be modeled as a process, where features are extracted from each source within a mixture, or, as "understanding without separation" [153]. Different representations are discussed in more detail in Section 1.4.

1.2 Applications

In most audio applications, applying some processing only to a certain source within a polyphonic mixture is virtually impossible. This creates a need for source separation methods, which first separate the mixture into sources, and then process the separated sources individually. Separation of sound sources has suggested to have applications, for example, in audio coding, analysis, and manipulation of audio signals.

Audio coding General-purpose audio codecs are typically based on perceptual audio coding, meaning that the objective is to quantize the signal so that the quantization errors are inaudible. Contrary to this, source coding aims at data reduction by utilizing redundancies in the data. Existing perceptual audio codecs enable a decent quality with all material, but source coding algorithms (particularly speech codecs) enable this by a much lower bit rate. While existing methods encode a polyphonic signal with a single stream, separating it into sources and then encoding each of them with a specific source codec could enable a higher coding efficiency and flexibility.

MPEG-4 [88, 137] is a state-of-the art audio and video coding standard by Moving Picture Experts Group (MPEG). Earlier MPEG standards are currently widely used in several consumer formats, for example mp3, DVD, and digital television. MPEG-4 includes two speech codecs [126, 127], both of which use a source model where the excitation by vocal cords and the filtering by the vocal track are modeled separately. For general audio coding it uses Advanced Audio Coding (AAC) [87], which is a perceptual audio codec. For low bit rates MPEG-4 uses a parametric codec [141], which represents the signal as a sum of individual sinusoids, harmonic tones, and noise. The representation of harmonic tones is similar to the representation used in the separation algorithm proposed in Chapter 6. Since the quality of existing separation algorithms is not sufficient for separating sources from complex polyphonic signals, material for MPEG-4 has to be produced so that each source is recorded as an individual track, so that separation is not needed.

Object-based coding refers to a technique where a signal is separated into sound objects, for example individual tones. This can be accomplished for example by grouping the elements of a parametric representation [183]. Object-based sound source modeling [174] aims at modeling the phenomenon, for example the musical instrument, that generated each object. This can be utilized in audio coding by transmitting only the estimated parameters, and synthesizing the sounds in the decoder. This is implemented in the MPEG-4 standard as

"Structured Audio" component [152], and the standard includes also a format for coding object-based audio scenes [160].

Analysis The analysis of polyphonic signals is difficult, since co-occurring sounds disturb the estimation. One approach towards solving this is to apply separation as a preprocessing step, and then analyze the separated signals. For example, fundamental frequency estimation of a separated monophonic instrument is a relatively easy task compared to the multiple fundamental frequency estimation of polyphonic music. Some of the methods presented in this thesis have already been applied successfully in automatic music transcription [136].

Manipulation Separation enables more efficient manipulation of music signals. A source can be removed, moved in time, or otherwise edited independently of the other sources. For example, material for music rehearsals and performances or karaoke applications could be produced by removing certain instruments or vocals from an existing recording. Currently these changes can be done only in the production stage where the unmixed signals for each sources are available. Processing of mixed signal with the existing tools enables only processing where a certain change is applied on all the concurrent sources.

Separation can also be viewed as noise suppression, where unwanted sources are separated and removed. A potential application of this is audio restoration [50], which aims at enhancing the quality of old recordings.

1.3 Approaches to One-Channel Sound Source Separation

The first works on one-channel sound source separation concentrated on the separation of speech signals [112, 134, 142, 202]. In the recent years, the analysis and processing of music signals has received an increasing amount of attention, which has also resulted in rapid development in music signal separation techniques.

Music is in some senses more challenging to separate than speech. Musical instruments have a wide range of sound production mechanisms, and the resulting signals have a wide range of spectral and temporal characteristics. Unlike speakers, who tend to pause when other speakers talk, sources constituting a musical piece often play simultaneously, and favor consonant pitch intervals. This increases the overlap of the sources significantly in a time-frequency domain, complicating their separation (see also Sections 1.4 and 5.3). Even though the acoustic signals are produced independently in each source, it is their consonance and interplay which makes up the music. This results in source signals which depend on each other, which may cause some separation criteria, such as statistical independence to fail.

Approaches used in one-channel sound source separation which do not use source-specific prior knowledge can be roughly divided into three categories,

which are model-based inference, unsupervised learning, and psychoacoustically motivated methods, which are shortly discussed as follows. In practice, many algorithms use principles from more than one category.

Model-based inference The methods in this category use a parametric model of the sources to be separated, and the model parameters are estimated from the observed mixture signal. Implicit prior information can be used to design deterministic, rule-based algorithms for the parameter estimation (for example [53, 119, 142]), or Bayesian estimation (for example [31, 38, 55]) can be used when the prior information is defined explicitly using probability density functions.

In music applications, the most commonly used parametric model is the sinusoidal model, which is discussed in more detail in Chapter 5. The model easily enables the prior information of harmonic spectral structure, which makes it most suitable for the separation of pitched musical instruments and voiced speech.

Unsupervised learning Unsupervised learning methods usually apply a simple non-parametric model, and use less prior information of the sources. Instead, they aim at learning the source characteristics from the data. The algorithms can apply information-theoretical principles, such as statistical independence between sources. Reducing the redundancy in the data has turned out to produce representations where the sources are present in isolation. Algorithms which are used to estimate the sources are based on independent component analysis (ICA), non-negative matrix factorization, and sparse coding. Chapter 2 introduces these algorithms and presents an overview of unsupervised learning methods in one-channel sound source separation. Our proposed unsupervised algorithms are described in Chapters 3 and 4.

Psychoacoustically motivated methods The cognitive ability of humans to perceive and recognize individual sound sources in a mixture referred to as auditory scene analysis [22]. Computational models of this function typically consist of two main stages so that an incoming signal is first decomposed into its elementary time-frequency components and these are then organized to their respective sound sources. Bregman [22] listed the following association cues in the organization:

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

These association cues have been used by several researchers [34, 47] to develop sound source separation algorithms. Later there has been criticism that the grouping rules can only describe the functioning the human hearing in simple

cases [163, p. 17], and robust separation of complex audio signals using them is difficult.

It is probable that the human auditory system uses both innate and learned principles in the separation [22, pp. 38-39], and the physiology of human peripheral auditory system explains some of the low-level innate mechanisms [204, p. 20], [40]. Even though also many higher-level segregation mechanisms can be assumed to be innate, the exact effect of learning is not known [22].

Even though our brain does not resynthesize the acoustic waveforms of each source separately, the human auditory system is a useful reference in the development of one-channel sound source separation systems, since it is the only existing system which can robustly separate sound sources in various circumstances.

Multi-channel methods Contrary to one-channel separation, multi-channel methods use recordings with multiple microphones placed at different positions. The main advantage of this is the availability of spatial information, which usually enables better separation quality than one-channel separation algorithms. When the acoustic sound waves travel from each source to each microphone, each source is delayed and attenuated, or filtered, differently. This enables recovering the sources by acoustic beamforming [46, pp. 293-311], or by blind separation of convolutive mixtures [177].

Acoustic beamforming attenuates or boosts a source signal depending on its direction of arrival, and the performance depends directly on the number of microphones. Blind separation of convolutive mixtures inverses the filtering and mixing process using algorithms extended from ICA. It enables theoretically a perfect separation when the number of microphones is equal to or larger than the number of sources. In practice, however, the movement of sources or microphones, or additional noise reduce the quality [176].

In the case of music signals, the one-channel separation principles have also been integrated into the multichannel separation framework, and this often increases the separation quality [59, 181, 196, 197, 205]. In the case of produced music, modeling the contribution of a source signal within a mixture by filtering with a fixed linear filter may not be valid, since at the production stage, nonlinear effects are often applied on the mixture signal.

1.4 Signal Representations

The most common digital representation of an acoustic signal is the sampled waveform, where each sample describes the sound pressure level of the signal at a particular time (see Fig. 1.1). Even though this representation allows many basic audio signal processing operations, it is not directly suitable for some more difficult tasks, such as separation. Therefore, the input signal is often represented using a so-called mid-level representation. As the word "to represent" suggest, the motivation for this is to bring out more clearly the

important characteristics of the signal for the application in hand. Different representations are illustrated in Figure 1.1.

Time-frequency representations Most methods use a time-frequency representation, where the input signal is divided into short *frames* (typically 10 - 100 ms in audio applications), windowed, and a frequency transform (typically the discrete Fourier transform, DFT, [21, pp. 356-384]) of each frame is taken. The frequency transform of a single frame is denoted by *spectrum*, and the magnitude of its each coefficient shows the energy at a particular frequency. The term *spectrogram* is used to denote the whole time-frequency representation, where the temporal locations of the frames determine the time axis.

Many perceptually important characteristics of a sound are determined by its spectrum [72], and also the human auditory system is known to perform frequency analysis of a certain kind [146, pp. 69-73]. The rough spectral energy distribution brings out the formant structure of a sound, which is unique for each instrument, and therefore an important cue in its identification. Spectral fine structure, on the other hand, reveals the vibration modes of the sound, which are often in harmonic relationships. This results in a clearly perceived *pitch*, which is a basis for tonal music. Most sounds contain also time-varying characteristics, which are represented by the time-varying spectrum.

Adaptive bases The short-time Fourier transform (STFT) represents each frame of an input signal as weighted sum of fixed basis functions, which are sinusoids and cosines of different frequencies. The basis functions can be as well estimated from the data. Ideally, the basis functions capture redundant characteristics of the data and therefore reduce the number of required basis functions. Algorithms that have been used to estimate the basis functions include, for example, independent component analysis [1] and various matching pursuit methods [35].

We can also first apply STFT and discard phases to obtain a phase-invariant representation. This can be further analyzed to obtain phase-invariant basis functions, each of which typically corresponds to a musically meaningful entity, for example an individual tone. These methods are discussed in more detail in Chapters 2 and 3.

Sparse representations Time-frequency representations of natural sounds typically have sparse distributions [10], meaning that most of the coefficients are approximately zero. Also the term *compact* has been used as a synonym for sparse [66, pp. 6-9]. The sparseness provides a good basis for separation: since only a few coefficients are non-zero, it is unlikely that two or more independent sources have a large coefficient in the same time-frequency point. Thus, estimating the most dominant source in each time-frequency point and then assigning all the energy at that point to the source often produces tolerable results. This separation method can be viewed as multiplication of the mixture spectrogram by a binary mask for each source. The binary masks can be estimated using all

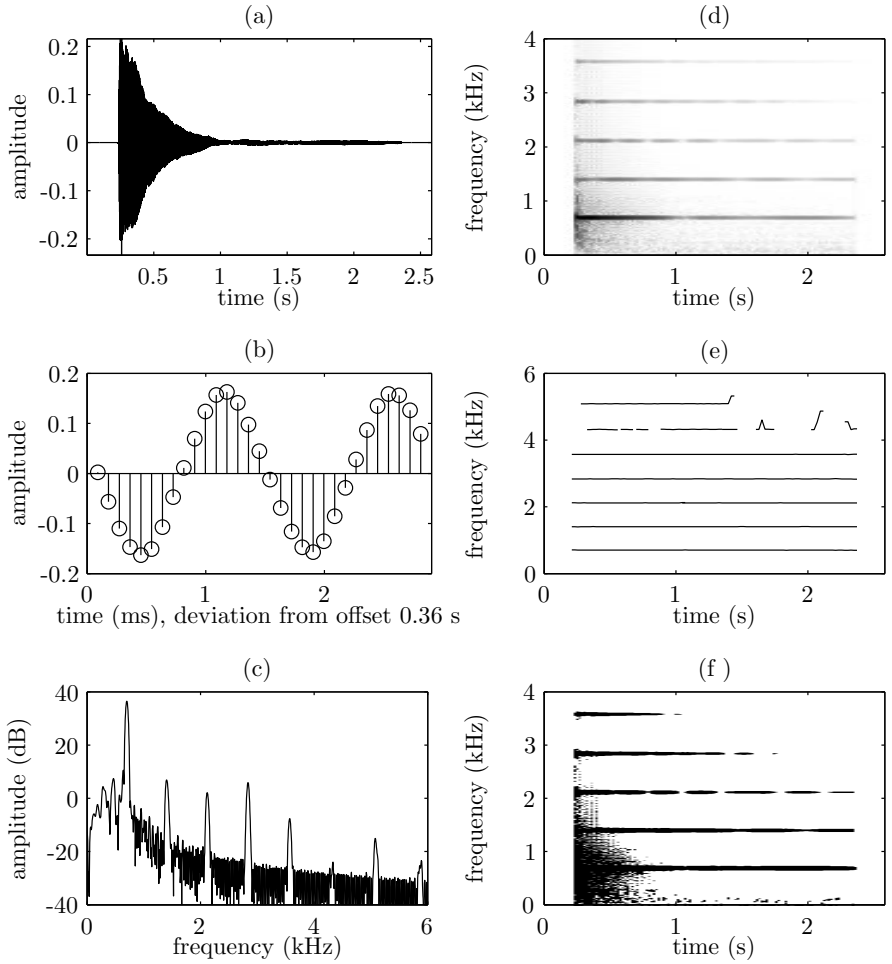

Figure 1.1: Representations of an acoustic signal of piano tone F5. On the left-hand column, the upper panel (a) shows the time-domain signal, the middle panel (b) a short excerpt of the same signal, so that individual samples can be seen, and (c) the amplitude spectrum of the signal. The right-hand column illustrates the spectrogram of the signal by different representations. The grey-scale spectrogram (d) illustrates the magnitude at each time-frequency point. The middle panel (e) shows the frequencies of a sinusoidal representation, where each partial is modeled using a sinusoidal trajectory. The bottom panel (f) shows active time-frequency points of a sparse representation, where only 2% of the original coefficients are active.

the principles described above, i.e., psychoacoustic [83], machine learning [147], or model-based [52] approaches. The quality can be further improved by using soft masks [144].

In the case of music signals, however, different instruments are more likely to have non-zero coefficients in the same time-frequency point. Most musical styles favor consonant intervals between different instruments, so that the ratio of their fundamental frequencies is a ratio of small integers, and therefore many of their vibrating modes have approximately equal frequencies. Furthermore, rhythmic congruity causes the activities to be temporally aligned. This results in a spectrum where a relatively large number of coefficients are active simultaneously. This makes the separation of music signals more challenging than speech separation.

Pitched musical instruments (and also speech) have typically either a continuous periodic excitation or damping of their harmonic modes is relatively slow, so that the coefficients corresponding to harmonic modes are temporally continuous. In the framework of sparse representations, the temporal continuity or harmonic relationships can be used to group the coefficients, which results in structured sparse representations [168]. These have been used, for example, in audio coding [36].

Parametric models Sparseness can also be used to develop parametric models for time-domain signals. The sinusoidal model used in Chapters 5 and 6 groups temporal trajectories of active frequency components into sinusoidal trajectories, which are parameterized by time-varying frequencies, amplitudes and phases. A more detailed explanation of the sinusoidal model is given in Chapter 5. Other parametric models for music signal include, for example, the transient model proposed by Verma [180, pp. 57-70].

The sinusoidal model, adaptive bases, and structured sparse representations fulfill most of the desirable properties of a mid-level representation suggested by Ellis in [49]: the representations are invertible so that an approximation of the original signals can be regenerated from them, they have a reduced number of components compared with the number of original samples, and each component is likely to represent a single sound source, to some degree.

Auditory mid-level representations Auditory mid-level representations, which model the signal processing of the human auditory system, received a lot of attention in the early research [49]. It can be assumed that by mimicking the human auditory system one could achieve the sound separation ability of humans. However, the exact mechanisms of higher-level signal processing which take place in the brain are not known. Therefore, this approach cannot be used to design an entire separation algorithm; however, it has been shown that the lower-level processing already accounts for some separation [40].

The signal processing of the peripheral auditory system can be modeled as a filter bank [203, pp. 43-44], followed by a cascade of half-wave rectification

and low-pass filtering of the subband signals, which accounts roughly to the mechanical-neural transduction of hair cell [122]. Later stages often include a periodicity analysis mechanism such as autocorrelation, but the exact mechanisms of this stage are not known. This processing leads to a three-dimensional *correlogram*, which represents the signal intensity as a function of time, frequency (filter bank channel), and autocorrelation lag. Many algorithms sum the channel-wise autocorrelations to result in a *summary autocorrelation function*, which provides a good basis for pitch estimation [122].

1.5 Quality Evaluation

A necessary condition for the development of source separation algorithms is the ability to measure the goodness of the results. In general, the separation quality can be measured by comparing separated signals with reference sources, or by listening to the separated signals. Reliable listening tests require a large number of listeners, and are therefore slow to conduct and expensive. Therefore, formal listening test have usually not been used in the quality evaluation of sound source separation algorithms, but objective computational measures are used.

In practice, quantitative evaluation of the separation quality requires that reference signals, i.e., the original signals before mixing, are available. Commercial music is usually produced so that instruments are recorded on individual tracks, which are later mixed. When target sources are individual instruments, these tracks could be used as references; however, this material is usually not available. Moreover, the mastering stage in music production often includes nonlinear effects on the mixture signal, so that the reference signals do not equal to the signals within the mixture any longer.

Many separation algorithms aim at separating individual tones of each instrument. Reference material where the tones are presented in isolation is difficult to record, and therefore synthesized material is often used. Generating test signals for this purpose is not a trivial task. For example, material generated using a software synthesizer may produce misleading results, since many synthesizers produce tones which are identical at each repetition.

The evaluation of an unsupervised learning based source separation system is especially difficult. Objective evaluation requires that reference sources are available, and results obtained with a set of reference signals are often used to develop the system. In practice this results in optimizing the system to a certain target material, making the system less "unsupervised". Furthermore, when it is not known which separated signal corresponds to which reference source, there has to be a method to associate the separated signals to their references. This can be done for example using the similarity between separated signals and the references, which again makes the the system less "unsupervised". These difficulties have also been discussed from a multi-channel blind source separation point of view in [156].

In the following, we shortly discuss some commonly used objective quality measures, which are here divided into three categories: low-level, perceptual,

and application-specific measures.

Low-level measures Low-level measures are simple statistics of the separated and reference signals. Many authors have used the *signal-to-distortion ratio* (SDR) as a measure to summarize the quality. It is the ratio of the energies of the reference signal and the error between the separated and reference signal, defined in decibels as

$$\text{SDR [dB]} = 10 \log_{10} \frac{\sum_n s(n)^2}{\sum_n [\hat{s}(n) - s(n)]^2}, \tag{1.2}$$

where $s(n)$ is a reference signal of the source before mixing, and $\hat{s}(n)$ is the separated signal. In the separation of music signals, Jang and Lee [90] reported average SDR of 9.6 dB for an algorithm which trains basis functions separately for each source. Helén and Virtanen [79] reported average SDR of 6.4 dB for their algorithm in the separation of drums and polyphonic harmonic track. Also the terms signal-to-noise or signal-to-residual ratio have often been used to refer to the SDR.

A simple objective measure which is perceptually more plausible is the segmental signal-to-distortion ratio [123], which is calculated as the average of frame-wise SDR's. Unlike the normal SDR, it takes into account the fact that errors in low-intensity segments are usually more easily perceived. The segmental SDR has often been used to measure the subjective quality of speech. Additional low-level performance measures for audio source separation tasks have been discussed, e.g., by Vincent et al. in [182]. For example, they measured the interference from other sources by the correlation of the separated signal to the other references.

Perceptual measures Perceptual measures in general estimate the audibility of the separation errors. Typically these process the reference and separated signal using an auditory model, and the difference between signals is calculated in the auditory domain [11–13]. Most perceptual measures are developed for the quality evaluation of coded speech and audio. They have usually been optimized for signals where the differences between the reference and target are caused by quantization. Therefore, they may produce misleading results when applied on separated signals, since typical errors in sound source separation are deletions, where a segment of a signal is an all-zero signal, and insertions, where separated signal contains interference from another sources, even though the reference signal is all-zero.

Application-specific measures In application-specific measures the separation accuracy is judged by the performance of the final application. For example, when the separation is used as preprocessing in a speech recognition system, the performance can be measured by the word-error rate [48], or when separation is used as preprocessing in automatic music transcription, the performance can be

measured by the transcription accuracy [136]. Unlike the low-level and perceptual measures, application-specific measures do not necessarily require reference signals.

1.6 Outline and main results of the thesis

The thesis outline is as follows: Chapter 2 gives an overview of unsupervised learning algorithms for sound source separation, which are a major class of methods applied in this thesis. Chapters 3 and 4 propose two unsupervised learning algorithms based on an instantaneous model and a convolutive model, respectively. Chapter 5 discusses a parametric sinusoidal model, which is used in the separation algorithm proposed in Chapter 6.

The original contributions of this thesis were published partly previously in [189–191, 193–195].[2] The author has also published other results related to the thesis in [185,186,188,192]. In addition, the research work has resulted in the collaborative publications [78, 79, 103, 104, 135, 136, 161]. The articles published in conference proceedings are available at `http://www.cs.tut.fi/~tuomasv/` in pdf format.

The main contributions of this thesis are the following:

- A unifying framework for the existing unsupervised learning algorithms based on the linear instantaneous model.
- A separation algorithm which combines non-negative matrix factorization with sparse coding and temporal continuity objectives. This includes a simple and efficient temporal continuity objective which increases the separation quality, and an augmented divergence objective, which is makes the optimization feasible with the projected steepest descent algorithm.
- A convolutive signal model which enables representing time-varying spectra and fundamental frequencies, and estimation algorithms which are based on minimization of the Euclidean distance or divergence with non-negativity restrictions.
- A separation algorithm based on a sinusoidal model, which includes a computationally efficient parameter estimation framework. This includes several methods for estimating overlapping harmonic partials by interpolating from the adjacent partials.

[2] The material from [193] is adapted to this thesis with kind permission, ©2006 Springer Science and Business Media LLC.

Chapter 2

Overview of Unsupervised Learning Methods for Source Separation

This chapter gives an overview of the existing unsupervised learning methods which have proven to produce applicable separation results in the case of music signals. Instead of sophisticated modeling of the source characteristics or the human auditory perception, they try to separate and learn source signal structures in mixed data based on information-theoretical principles, such as statistical independence between sources. Algorithms which implement the learning are *independent component analysis* (ICA), *sparse coding*, and *non-negative matrix factorization* (NMF), which have been recently used in machine learning tasks in several application areas. Although the motivation of unsupervised learning algorithms is not in the human auditory perception, they can lead to similar processing. For example, all the unsupervised learning methods discussed here lead to reducing redundancy in data, and it has been found that redundancy reduction takes place in the auditory pathway, too [32].

All the algorithms mentioned above (ICA, sparse coding, and NMF) can be formulated using a linear signal model which is explained in Section 2.1. When used for monaural audio source separation, these algorithms usually factorize the spectrogram or other short-time representation of the input signal into elementary components. Different data representations in this framework are discussed in Section 2.1.2 and estimation criteria and algorithms are discussed in Sections 2.2, 2.3, and 2.4. Methods for obtaining and utilizing prior information are presented in Section 2.5. Once the input signal is factorized into components, the components can be clustered into sources, analyzed to obtain musically important information, or synthesized, as discussed in Section 2.6. The described algorithms are evaluated and compared in Section 3.3.3 in the next chapter.

2.1 Linear Signal Model

When several sound sources are present simultaneously, the acoustic waveform of the mixture signal is a linear superposition of the source waveforms. Many unsupervised learning algorithms, for example the standard ICA, require that the number of sensors is larger or equal to the number of sources in order that the separation be possible. In multichannel sound separation, this means that there should be at least as many microphones as there are sources. However, automatic transcription of music usually aims at finding the notes in monaural (or stereo) signals, for which basic ICA methods cannot be used directly. By using a suitable signal representation, the methods become applicable with one-channel data.

The most common representation of monaural signals is based on short-time signal processing, in which the input signal is divided into (possibly overlapping) frames. Frame sizes between 20 and 100 ms are typical in systems which aim at the separation of musical signals. Some systems operate directly on time-domain signals and some others take a frequency transform, for example the DFT of each frame.

2.1.1 Basis Functions and Gains

The representation of the input signal within each frame $t = 1 \dots T$ is denoted by an observation vector $\mathbf{x}_t$. The methods discussed in this chapter model $\mathbf{x}_t$ as a weighted sum of basis functions $\mathbf{b}_j$, $j = 1 \dots J$, so that the signal model $\hat{\mathbf{x}}_t$ can be written as

$$\hat{\mathbf{x}}_t = \sum_{j=1}^{J} g_{j,t} \mathbf{b}_j \qquad t = 1, \dots, T, \tag{2.1}$$

where J is the number of basis functions, and $g_{j,t}$ is the gain of the j^{th} basis function in the t^{th} frame. Some methods estimate both the basis functions and the time-varying gains from a mixed input signal, whereas others use pre-trained basis functions or some prior information about the gains.

The term *component* refers to one basis function together with its time-varying gain. Each sound source is modeled as a sum of one or more components, so that the model for source m in frame t is written as

$$\hat{\mathbf{y}}_{m,t} = \sum_{j \in S_m} g_{j,t} \mathbf{b}_j, \tag{2.2}$$

where S_m is the set of components within source m. The sets are disjoint, i.e., each component belongs to one source only.

An *observation matrix* $\mathbf{X} = \left[\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_T\right]$ is used to represent the observations within T frames. The model (2.1) can be written in a matrix form as

$$\hat{\mathbf{X}} = \mathbf{B}\mathbf{G}, \tag{2.3}$$

where $\mathbf{B} = \left[\mathbf{b}_1, \mathbf{b}_2, \dots, \mathbf{b}_J\right]$ is the *basis matrix*, and $[\mathbf{G}]_{j,t} = g_{j,t}$ is the *gain matrix*. The notation $[\mathbf{G}]_{j,t}$ is used to denote the $(j,t)^{\text{th}}$ entry of matrix $\mathbf{G}$.

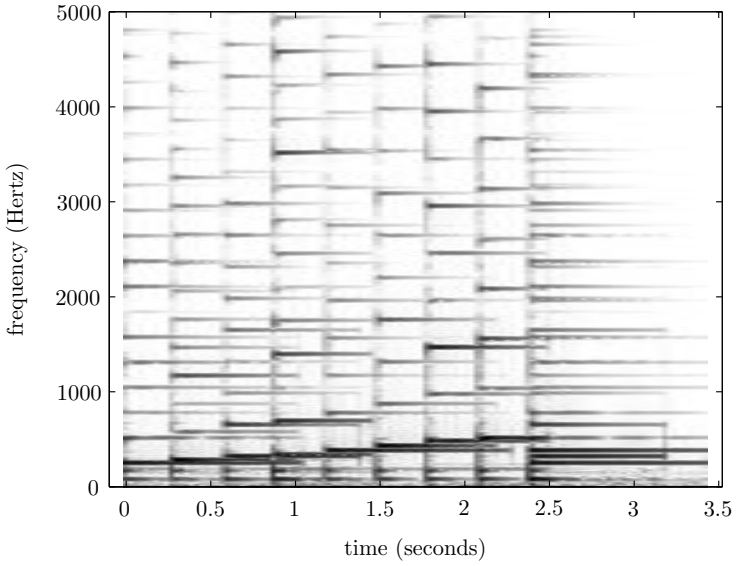

Figure 2.1: Spectrogram of a signal which consist of a diatonic scale from C5 to C6, followed by a C major chord (simultaneous tones C5, E4, and G5), played by an acoustic guitar. The tones are not damped, meaning that consecutive tones overlap with each other.

The estimation algorithms can be used with several data representations. Often the absolute values of the DFT are used, which is referred to as *magnitude spectrum* in the following. In this case, $\mathbf{x}_t$ is the magnitude spectrum within frame t, and each component j has a fixed magnitude spectrum $\mathbf{b}_j$ with a time-varying gain $g_{j,t}$. The observation matrix consisting of frame-wise magnitude spectra is here called a *magnitude spectrogram*. Other representations are discussed in Section 2.1.2.

The model (2.1) is flexible in the sense that it is suitable for representing both harmonic and percussive sounds. It has been successfully used in the transcription of drum patterns [58, 136], in the pitch estimation of speech signals [159], and in the analysis of polyphonic music signals [4, 18, 30, 115, 166, 178, 184, 189].

Fig. 2.1 shows an example signal which consists of a diatonic scale and a C major chord played by an acoustic guitar. The signal was separated into components using the NMF algorithm that will be described in Section 2.4, and the resulting components are depicted in Fig. 2.2. Each component corresponds roughly to one fundamental frequency: the basis functions are approximately harmonic spectra and the time-varying gains follow the amplitude envelopes of the tones. The separation is not perfect because of estimation inaccuracies. For example, in some cases the gain of a decaying tone drops to zero when a new tone begins.

Factorization of the spectrogram into components with a fixed spectrum and

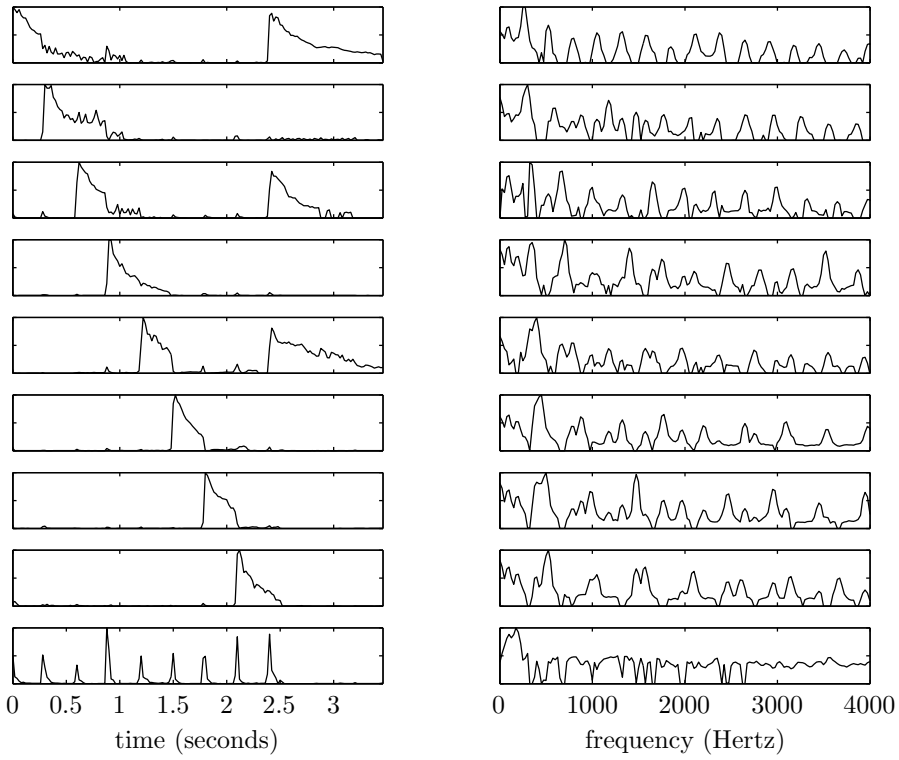

Figure 2.2: Components estimated from the example signal in Fig. 2.1. Basis functions are plotted on the right and the corresponding time-varying gains on the left. Each component except the undermost one corresponds to an individual pitch value and the gains follow roughly the amplitude envelope of each tone. The undermost component models the attack transients of the tones. The components were estimated using the NMF algorithm [114, 166] and the divergence objective (explained in Section 2.4).

a time-varying gain has been adopted as a part of the MPEG-7 pattern recognition framework [29], where the basis functions and the gains are used as features for classification. Kim et al. [98] compared these to Mel-frequency cepstral coefficients (MFCCs) which are commonly used features in the classification of audio signals. In this study, MFCCs performed better in the recognition of sound effects and speech than features based on ICA or NMF. However, final conclusions about the applicability of these methods to sound source recognition are yet to be made. The spectral basis decomposition specified in MPEG-7 models the summation of components on a decibel scale, which makes it unlikely that the separated components correspond to physical sound objects.

2.1.2 Data Representation

The model (2.1) presented in the previous section can be used with time-domain or frequency-domain observations and basis functions. Time-domain observation vector $\mathbf{x}_t$ is the signal within frame t multiplied by the window function w:

$$\mathbf{x}_t = \big[x(n_t)w(0),\, x(n_t+1)w(1),\, \ldots,\, x(n_t+N-1)w(N-1)\big]^\mathsf{T}, \qquad (2.4)$$

where n_t is the index of the first sample of the t^{th} frame. A frequency-domain observation vector is obtained by applying a chosen frequency transformation, such as DFT, on the time-domain vector. The representation of the signal and the basis functions have to be the same. ICA and sparse coding allow the use of any short-time signal representation, whereas for NMF, only frequency-domain representations allowing non-negativity restrictions are appropriate. Naturally, the representation has a significant effect on performance. The advantages and disadvantages of different representations are considered in this section. For a more extensive discussion, see Casey [28] or Smaragdis [163].

Time-domain Representation Time-domain representations are straightforward to compute, and all the information is preserved when an input signal is segmented into frames. However, time-domain basis functions are problematic in the sense that a single basis function alone cannot represent a meaningful sound source: the phase of the signal within each frame varies depending on the frame position. In the case of a short-duration percussive source, for example, a separate basis function is needed for every possible position of the sound event within the frame. A shift-invariant model which is later discussed in Chapter 4 is one possible approach to overcome this limitation [18].

The time-domain signals of real-world sound sources are generally not identical at different occurrences since the phases behave very irregularly. For example, the overtones of a pitched musical instrument are not necessarily phase-locked, so that the time-domain waveform varies over time. Therefore, one has to use multiple components to represent even a single tone of a pitched instrument. In the case of percussive sound sources, this phenomenon is even clearer: the time-domain waveforms vary a lot at different occurrences even when their power spectra are quite similar.

The larger the number of the components, the more uncertain is their estimation and further analysis, and the more observations are needed. If the sound event represented by a component occurs only once in the input signal, separating it from co-occurring sources is difficult since there is no information about the component elsewhere in the signal. Also, clustering the components into sources becomes more difficult when there are many of them for each source.

Separation algorithms which operate on time-domain signals have been proposed for example by Dubnov [45], Jang and Lee [90], and Blumensath and Davies [18]. Abdallah and Plumbley [1, 2] found that the independent components analyzed from time-domain music and speech signals were similar to a wavelet or short-time DFT basis. They trained the basis functions using several days of radio output from BBC Radio 3 and 4 stations.

Frequency-domain Representation The phases of a signal can be discarded by taking a frequency transform, such as the DFT, and considering only the magnitude or power spectrum. Even though some information is lost, this also eliminates the above-discussed phase-related problems of time-domain representations. Also the human auditory perception is quite insensitive to phase. Contrary to time-domain basis functions, many real-world sounds can be rather well approximated with a fixed magnitude spectrum and a time-varying gain, as seen in Figs. 2.1 and 2.2, for example. Sustained instruments in particular tend to have quite a stationary spectrum after the attack transient.

In most systems aimed at the separation of sound sources, DFT and a fixed window size is applied, but the estimation algorithms allow the use of any time-frequency representation. For example, a logarithmic spacing of frequency bins has been used [25], which is perceptually and musically more plausible than a constant spectral resolution.

Two time-domain signals of concurrent sounds and their complex-valued DFTs $Y_1(k)$ and $Y_2(k)$ sum linearly, $X(k) = Y_1(k) + Y_2(k)$. This equality does not apply for their magnitude or power spectra. However, provided that the phases of $Y_1(k)$ and $Y_2(k)$ are uniformly distributed and independent of each other, we can write

$$E\{|X(k)|^2\} = |Y_1(k)|^2 + |Y_2(k)|^2, \tag{2.5}$$

where $E\{\cdot\}$ denotes expectation. This means that in the expectation sense, we can approximate time-domain summation in the power spectral domain, a result which holds for more than two sources as well. Even though magnitude spectrogram representation has been widely used and it often produces good results, it does not have a similar theoretical justification.

Since the summation of power or magnitude spectra is not exact, use of phaseless basis functions causes an additional source of error. The phase spectra of natural sounds are very unpredictable, and therefore the separation is usually done using a phaseless representation, and if a time-domain signals of the separated sources are required, the phases are generated afterwards. Methods for phase generation are discussed in Section 2.6.3.

The human auditory system has a wide dynamic range: the difference between the threshold of hearing and the threshold of pain is approximately 100 dB [146]. Unsupervised learning algorithms tend to be more sensitive to high-energy observations. If sources are estimated from the power spectrum, some methods fail to separate low-energy sources even though they would be perceptually and musically meaningful. This problem has been noticed, e.g., by FitzGerald in the case of percussive source separation [56, pp. 93-100]. To overcome the problem, he used an algorithm which processed separately high-frequency bands which contain low-energy sources, such as hi-hats and cymbals [57]. Also Vincent and Rodet [184] addressed the same problem. They proposed a model in which the noise was modeled to be additive in the log-spectral domain. The numerical range of a logarithmic spectrum is compressed, which increases the sensitivity to low-energy sources. Additive noise in the log-spectral domain corresponds to multiplicative noise in power spectral domain, which was also assumed in the system proposed by Abdallah and Plumbley [4]. Virtanen proposed the use of perceptually motivated weights [190]. He used a weighted cost function, in which the observations were weighted so that the quantitative significance of the signal within each critical band was equal to its contribution to the total loudness.

2.2 Independent Component Analysis

ICA has been successfully used in several blind source separation tasks, where very little or no prior information is available about the source signals. One of its original target applications was multichannel sound source separation, but it has also had several other uses. ICA attempts to separate sources by identifying latent signals that are maximally independent. In practice, this usually leads to the separation of meaningful sound sources.

Mathematically, statistical independence is defined in terms of probability densities: random variables x and y are said to be independent if their joint probability distribution function $\mathsf{p}(x, y)$ is a product of the marginal distribution functions, $\mathsf{p}(x, y) = \mathsf{p}(x)\mathsf{p}(y)$ [80, pp. 23-31, 80-89].

The dependence between two variables can be measured in several ways. Mutual information is a measure of the information that given random variables have on some other random variables [86]. The dependence is also closely related to the Gaussianity of the distribution of the variables. According to the central limit theorem, the distribution of the sum of independent variables is more Gaussian than their original distributions, under certain conditions. Therefore, some ICA algorithms aim at separating output variables whose distributions are as far from Gaussian as possible.

The signal model in ICA is linear: K observed variables $x_1, \ldots, x_K$ are modeled as linear combinations of J source variables $g_1, \ldots, g_J$. In a vector-matrix form, this can be written as

$$\mathbf{x} = \mathbf{B}\mathbf{g}, \tag{2.6}$$

where $\mathbf{x} = \left[x_1, \ldots x_K\right]^\mathsf{T}$ is an observation vector, $[\mathbf{B}]_{k,j} = b_{k,j}$ is a mixing matrix, and $\mathbf{g} = \left[g_1, \ldots, g_J\right]^\mathsf{T}$ is a source vector. Both $\mathbf{B}$ and $\mathbf{g}$ are unknown.

The standard ICA requires that the number of observed variables K (the number of sensors), is equal to the number of sources J. In practice, the number of sensors can also be larger than the number of sources, because the variables are typically decorrelated using principal component analysis (PCA, [33, pp. 183-186]), and if the desired number of sources is less than the number of variables, only the desired number of principal components with the largest energy are selected.

As another preprocessing step, the observed variables are usually centered by subtracting their mean and by normalizing their variance to unity. The centered and whitened data observation vector $\mathbf{x}$ is obtained from the original observation vector $\tilde{\mathbf{x}}$ by

$$\mathbf{x} = \mathbf{V}(\tilde{\mathbf{x}} - \boldsymbol{\mu}), \tag{2.7}$$

where $\boldsymbol{\mu}$ is the empirical mean of the observation vector, and $\mathbf{V}$ is a whitening matrix, which is often obtained from the eigenvalue decomposition of the empirical covariance matrix [80, pp. 408-409] of the observations [86].

To simplify the notation, it is assumed that the data $\mathbf{x}$ in (2.6) is already centered and decorrelated, so that $K = J$. The core ICA algorithm carries out the estimation of an unmixing matrix $\mathbf{W} \approx \mathbf{B}^{-1}$, assuming that $\mathbf{B}$ is invertible. Independent components are obtained by multiplying the whitened observations by the estimate of the unmixing matrix, to result in the source vector estimate $\hat{\mathbf{g}}$

$$\hat{\mathbf{g}} = \mathbf{W}\mathbf{x}. \tag{2.8}$$

The matrix $\mathbf{W}$ is estimated so that the output variables, i.e., the elements of $\hat{\mathbf{g}}$, become maximally independent. There are several criteria and algorithms for achieving this. The criteria, such as nongaussianity and mutual information, are usually measured using high-order cumulants such as kurtosis, or expectations of other nonquadratic functions [86]. ICA can be also viewed as an extension of PCA. The basic PCA decorrelates variables so that they are independent up to second-order statistics. It can be shown that if the variables are uncorrelated after taking a suitable non-linear function, the higher-order statistics of the original variables are independent, too. Thus, ICA can be viewed as a nonlinear decorrelation method.

Compared with the previously presented linear model (2.1), the standard ICA model (2.6) is exact, i.e., $\hat{\mathbf{x}} = \mathbf{x}$. Some special techniques can be used in the case of a noisy signal model, but often noise is just considered as an additional source variable. Because of the dimension reduction with PCA, $\mathbf{Bg}$ gives an exact model for the PCA-transformed observations but not necessarily for the original ones.

There are several ICA algorithms, and some implementations are freely available, such as FastICA [54, 84] and JADE [27]. Computationally quite efficient separation algorithms can be implemented based on FastICA, for example.

2.2.1 Independent Subspace Analysis

The idea of *independent subspace analysis* (ISA) was originally proposed by Hyvärinen and Hoyer [85]. It combines the multidimensional ICA with invariant feature extraction, which are shortly explained later in this section. After the work of Casey and Westner [30], the term ISA has been commonly used to denote techniques which apply ICA to factorize the spectrogram of a monaural audio signal to separate sound sources. ISA provides a theoretical framework for the whole separation algorithm discussed in this chapter, including spectrogram representation, decomposition by ICA, and clustering. Some authors use the term ISA also to refer to methods where some other algorithm than ICA is used for the factorization [184].

The general ISA procedure consists of the following steps:

1. Calculate the magnitude spectrogram $\tilde{\mathbf{X}}$ (or some other representation) of the input signal

2. Apply PCA[1] on the matrix $\tilde{\mathbf{X}}$ of size $(K \times T)$ to estimate the number of components J and to obtain whitening and dewhitening matrices $\mathbf{V}$ and $\mathbf{V}^+$, respectively. Centered, decorrelated and dimensionally-reduced observation matrix $\mathbf{X}$ of size $(J \times T)$ is obtained as $\mathbf{X} = \mathbf{V}(\tilde{\mathbf{X}} - \boldsymbol{\mu}\mathbf{1}^\mathsf{T})$, where $\mathbf{1}$ is an all-one vector of length T.

3. Apply ICA to estimate an unmixing matrix $\mathbf{W}$. The matrices $\mathbf{B}$ and $\mathbf{G}$ are obtained as $\mathbf{B} = \mathbf{W}^{-1}$ and $\mathbf{G} = \mathbf{W}\mathbf{X}$.

4. Inverse the decorrelation operation in Step 2 in order to get the mixing matrix $\tilde{\mathbf{B}} = \mathbf{V}^+\mathbf{B}$ and source matrix $\tilde{\mathbf{G}} = \mathbf{G} + \mathbf{W}\mathbf{V}\boldsymbol{\mu}\mathbf{1}^\mathsf{T}$ for the original observations $\tilde{\mathbf{X}}$.

5. Cluster the components to sources (see Section 2.6.1).

The motivation for above steps is given below. Depending on the application, all of them are not necessarily needed. For example, prior information can be used to set the number of components in Step 2.

The basic ICA is not directly suitable for the separation of one-channel signals, since the number of sensors has to be larger than or equal to the number of sources. Short-time signal processing can be used in an attempt to overcome this limitation. Taking a frequency transform such as DFT, each frequency bin can be considered as a sensor which produces an observation in each frame. With the standard linear ICA model (2.6), the signal is modeled as a sum of components, each of which has a static spectrum (or some other basis function) and a time-varying gain.

The spectrogram factorization has its motivation in invariant feature extraction, which is a technique proposed by Kohonen [107]. The short-time spectrum can be viewed as a set of features calculated from the input signal. As discussed in Section 2.1.2, it is often desirable to have shift-invariant basis functions, such

[1] Also singular value decomposition can be used to estimate the number of components [30].

as the magnitude or power spectrum [85,107]. Multidimensional ICA (explained below) is used to separate phase-invariant features into *invariant feature subspaces*, where each source is modeled as the sum of one or more components [85].

Multidimensional ICA [26] is based on the same linear generative model (2.6) as ICA, but the components are not assumed to be mutually independent. Instead, it is assumed that the components can be divided into disjoint sets, so that the components within each set may be dependent on each other, while dependencies between sets are not allowed. One approach to estimate multidimensional independent components is to first apply standard ICA to estimate the components, and then group them into sets by measuring dependencies between them.[2]

ICA algorithms aim at maximizing the independence of the elements of the source vector $\hat{\mathbf{g}} = \mathbf{W}\mathbf{x}$. In ISA, the elements correspond to the time-varying gains of each component. However, the objective can also be the independence of the spectra of components, since the roles of the mixing matrix and gain matrix can be swapped by $\mathbf{X} = \mathbf{B}\mathbf{G} \Leftrightarrow \mathbf{X}^\mathsf{T} = \mathbf{G}^\mathsf{T}\mathbf{B}^\mathsf{T}$. The independence of both the time-varying gains and basis functions can be obtained by using the spatiotemporal ICA algorithm [172]. There are not exhaustive studies regarding different independence criteria in monaural audio source separation. Smaragdis argued that in the separation of complex sources, the criterion of independent time-varying gains is better, because of the absence of consistent spectral characters [163]. FitzGerald reported that the spatiotemporal ICA did not produce significantly better results than normal ICA which assumes the independence of gains or spectra [56].

The number of frequency channels is usually larger than the number of components to be estimated with ICA. PCA or singular value decomposition (SVD) of the spectrogram can be used to estimate the number of components automatically. The components with the largest singular values are chosen so that the sum of their singular values is larger than or equal to a pre-defined threshold $0 < \theta \leq 1$ [30].

ISA has been used for general audio separation by Casey and Westner [30], for the analysis of musical trills by Brown and Smaragdis [25], and for percussion transcription by Fitzgerald et al. [57], to mention some examples. Also, a sound recognition system based on ISA has been adopted in the MPEG-7 standardization framework [29].

2.2.2 Non-Negativity Restrictions

When magnitude or power spectrograms are used, the basis functions are magnitude or power spectra which are non-negative by definition. Therefore, it can be advantageous to restrict the basis functions to be entry-wise non-negative. Also, it may be useful not to allow negative gains, but to constrain the components to be purely additive. Standard ICA is problematic in the sense that

[2]ICA aims at maximizing the independence of the output variables, but it cannot guarantee their complete independence, as this depends also on the input signal.

it does not enable these constraints. In practice, ICA algorithms also produce negative values for the basis functions and gains, and often there is no physical interpretation for such components.

ICA with non-negativity restrictions has been studied for example by Plumbley and Oja [139], and the topic is currently under active research. Existing non-negative ICA algorithms can enforce non-negativity for the latent variable matrix $\mathbf{G}$ but not for the mixing matrix $\mathbf{B}$. They also assume that the probability distribution of the source variables g_j is nonzero all the way down to zero, i.e., the probability $g_j < \delta$ is non-zero for any $\delta > 0$. This assumption may not hold in the case of some sound sources, which prevents the separation. Furthermore, the algorithms are based on a noise-free mixing model and in our experiments with audio spectrograms, they tended to be rather sensitive to noise.

It has turned out that the non-negativity restrictions alone are sufficient for the separation of the sources, without the explicit assumption of statistical independence. They can be implemented, for example, using the NMF algorithms discussed in Section 2.4.

2.3 Sparse Coding

Sparse coding represents a mixture signal in terms of a small number of active elements chosen out of a larger set [130]. This is an efficient approach for learning structures and separating sources from mixed data. In the linear signal model (2.3), the sparseness restriction is usually applied on the gains $\mathbf{G}$, which means that the probability of an element of $\mathbf{G}$ being zero is high. As a result, only a few components are active at a time and each component is active only in a small number of frames. In musical signals, a component can represent, e.g., all the equal-pitched tones of an instrument. It is likely that only a small number of pitches are played simultaneously, so that the physical system behind the observations generates sparse components.

In this section, a probabilistic framework is presented, where the source and mixing matrices are estimated by maximizing their posterior distributions. The framework is similar to the one presented by Olshausen and Field [130]. Several assumptions of, e.g., the noise distribution and prior distribution of the gains are used. Obviously, different results are obtained by using different distributions, but the basic idea is the same. The method presented here is also closely related to the algorithms proposed by Abdallah and Plumbley [3] and Virtanen [189], which were used in the analysis of music signals.

The posterior distribution [80, p. 228] of $\mathbf{B}$ and $\mathbf{G}$ given an observed spectrogram $\mathbf{X}$ is denoted by $\mathsf{p}(\mathbf{B}, \mathbf{G}|\mathbf{X})$. Based on Bayes' formula, the maximization of this can be formulated as [97, p. 351]

$$\max_{\mathbf{B},\mathbf{G}} \mathsf{p}(\mathbf{B}, \mathbf{G}|\mathbf{X}) \propto \max_{\mathbf{B},\mathbf{G}} \mathsf{p}(\mathbf{X}|\mathbf{B}, \mathbf{G})\mathsf{p}(\mathbf{B}, \mathbf{G}), \tag{2.9}$$

where $\mathsf{p}(\mathbf{X}|\mathbf{B}, \mathbf{G})$ is the probability of observing $\mathbf{X}$ given $\mathbf{B}$ and $\mathbf{G}$, and $\mathsf{p}(\mathbf{B}, \mathbf{G})$ is the joint prior distribution of $\mathbf{B}$ and $\mathbf{G}$.

For mathematical tractability, it is typically assumed that the noise (the residual $\hat{\mathbf{X}} - \mathbf{X}$) is i.i.d., independent from the model $\mathbf{BG}$, and normally distributed with variance σ^2 and zero mean. The likelihood of $\mathbf{B}$ and $\mathbf{G}$ can then be written as

$$\mathsf{p}(\mathbf{X}|\mathbf{B},\mathbf{G}) = \prod_{t,k} \frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{([\mathbf{X}]_{k,t} - [\mathbf{BG}]_{k,t})^2}{2\sigma^2}\right). \tag{2.10}$$

It is further assumed here that $\mathbf{B}$ has a uniform prior, so that $\mathsf{p}(\mathbf{B},\mathbf{G}) \propto \mathsf{p}(\mathbf{G})$. Each time-varying gain $[\mathbf{G}]_{j,t}$ is assumed to have a sparse probability distribution function of the exponential form

$$\mathsf{p}([\mathbf{G}]_{j,t}) = \frac{1}{Z} \exp\left(-f([\mathbf{G}]_{j,t})\right). \tag{2.11}$$

A normalization factor Z has to be used so that the density function sums to unity. The function f is used to control the shape of the distribution and is chosen so that the distribution is uni-modal and peaked at zero with heavy tails. Some examples are given later.

For simplicity, all the entries of $\mathbf{G}$ are assumed to be independent from each other, so that the probability distribution function of $\mathbf{G}$ can be written as a product of the marginal densities:

$$\mathsf{p}(\mathbf{G}) = \prod_{j,t} \frac{1}{Z} \exp\left(-f([\mathbf{G}]_{j,t})\right). \tag{2.12}$$

It is obvious that in practice the gains are not independent of each other, but this approximation is done to simplify the calculations. From the above definitions we get

$$\begin{aligned}\max_{\mathbf{B},\mathbf{G}} \mathsf{p}(\mathbf{B},\mathbf{G}|\mathbf{X}) \propto \max_{\mathbf{B},\mathbf{G}} \prod_{t,k} \frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{([\mathbf{X}]_{k,t} - [\mathbf{BG}]_{k,t})^2}{2\sigma^2}\right) \\ \times \prod_{j,t} \frac{1}{Z} \exp\left(-f([\mathbf{G}]_{j,t})\right).\end{aligned} \tag{2.13}$$

By taking a logarithm, the products become summations, and the exp-operators and scaling terms can be discarded. This can be done since logarithm is order-preserving and therefore does not affect the maximization. The sign is changed to obtain a minimization problem

$$\min_{\mathbf{B},\mathbf{G}} \sum_{t,k} \frac{([\mathbf{X}]_{k,t} - [\mathbf{BG}]_{k,t})^2}{2\sigma^2} + \sum_{j,t} f([\mathbf{G}]_{j,t}) \tag{2.14}$$

which can be written as

$$\min_{\mathbf{B},\mathbf{G}} \frac{1}{2\sigma^2}||\mathbf{X} - \mathbf{BG}||_F^2 + \sum_{j,t} f([\mathbf{G}]_{j,t}), \tag{2.15}$$

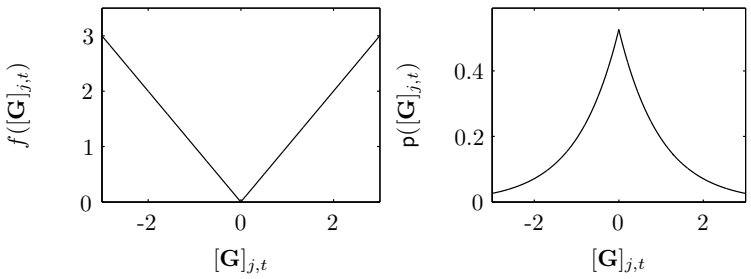

Figure 2.3: The cost function $f(x) = |x|$ (left) and the corresponding Laplacian prior distribution $\mathsf{p}(x) = \frac{1}{2}\exp(-|x|)$ (right). Values of $\mathbf{G}$ near zero are given a smaller cost and a higher probability.

where the Frobenius norm of a matrix is defined as

$$||\mathbf{Y}||_F = \sqrt{\sum_{i,j} [\mathbf{Y}]_{i,j}^2}. \tag{2.16}$$

In (2.15), the function f is used to penalize "active" (non-zero) entries of $\mathbf{G}$. For example, Olshausen and Field [130] suggested the functions $f(x) = \log(1 + x^2)$, $f(x) = |x|$, and $f(x) = x^2$. In audio source separation, Benaroya et al. [15] and Virtanen [189] have used $f(x) = |x|$. The prior distribution used by Abdallah and Plumbley [1, 3] corresponds to the function

$$f(x) = \begin{cases} |x|, & |x| \geq \mu \\ \mu(1-\alpha) + \alpha|x|, & |x| < \mu \end{cases}, \tag{2.17}$$

where the parameters μ and α control the relative mass of the central peak in the prior, and the term $\mu(1-\alpha)$ is used to make the function continuous at $x = \pm\mu$. All these functions give a smaller cost and a higher prior probability for gains near zero. The cost function $f(x) = |x|$ and the corresponding Laplacian prior $\mathsf{p}(x) = \frac{1}{2}\exp(-|x|)$ are illustrated in Fig. 2.3.

From (2.15) and the above definitions of f, it can be seen that a sparse representation is obtained by minimizing a cost function which is the weighted sum of the reconstruction error term $||\mathbf{X} - \mathbf{BG}||_F^2$ and the term which incurs a penalty on non-zero elements of $\mathbf{G}$. The variance σ^2 is used to balance between these two.

Typically, f increases monotonically as a function of the absolute value of its argument. The presented objective requires that the scale of either the basis functions or the gains are somehow fixed. Otherwise, the second term in (2.15) could be minimized without affecting the first term by setting $\mathbf{B} \leftarrow \mathbf{B}\theta$ and $\mathbf{G} \leftarrow \mathbf{G}/\theta$, where the scalar $\theta \to \infty$. The scale of the basis functions can be fixed for example with an additional constraint $||\mathbf{b}_j|| = 1$ as done by Hoyer [81], or the variance of the gains can be fixed.

The minimization problem (2.15) is usually solved using iterative algorithms. If both $\mathbf{B}$ and $\mathbf{G}$ are unknown, the cost function may have several local minima, and in practice reaching the global optimum in a limited time cannot be guaranteed. Standard optimization techniques based on steepest descent, covariant gradient, quasi-Newton, and active-set methods can be used. Different algorithms and objectives are discussed for example by Kreutz-Delgado et al. in [108]. Our proposed method is presented in Chapter 3. If $\mathbf{B}$ is fixed, more efficient optimization algorithms can be used. This can be the case for example when $\mathbf{B}$ is learned in advance from a training material where sounds are presented in isolation. These methods are discussed in Section 2.5.

No methods have been proposed for estimating the number of sparse components in a monaural audio signal. Therefore, J has to be set either manually, using some prior information, or to a value which is clearly larger than the expected number of sources. It is also possible to try different numbers of components and to determine a suitable value of J from the outcome of the trials.

As discussed in the previous section, non-negativity restrictions can be used for frequency-domain basis functions. With a sparse prior and non-negativity restrictions, one can use, for example, projected steepest descent algorithms which are discussed, e.g., by Bertsekas in [16, pp. 203-224]. Hoyer [81, 82] proposed a non-negative sparse coding algorithm by combining NMF and sparse coding. His algorithm used a multiplicative rule to update $\mathbf{B}$, and projected steepest descent to update $\mathbf{G}$.

In musical signal analysis, sparse coding has been used for example by Abdallah and Plumbley [3,4] to produce an approximate piano-roll transcription of synthesized harpsichord music, by Benaroya, McDonagh, Bimbot, and Gribonval to separate two pre-trained sources [15], and by Virtanen [189] to transcribe drums in polyphonic music signals synthesized from MIDI. Also, Blumensath and Davies used a sparse prior for the gains, even though their system was based on a different signal model [18].

2.4 Non-Negative Matrix Factorization

As discussed in Section 2.2.2, it is reasonable to restrict frequency-domain basis functions and their gains to non-negative values. As noticed by Lee and Seung [113], the non-negativity restrictions can be efficient in learning representations where the whole is represented as a combination of parts which have an intuitive interpretation.

The spectrograms of musical signals often have a unique decomposition into non-negative components, each of which represents parts of a single sound source. Therefore, in the signal model $\mathbf{X} \approx \mathbf{BG}$ the element-wise non-negativity of $\mathbf{B}$ and $\mathbf{G}$ alone is a sufficient condition for the separation of sources in many cases, without an explicit assumption of the independence of the sources.

Paatero and Tatter proposed a NMF algorithm in which the weighted energy of the residual matrix $\mathbf{X} - \mathbf{BG}$ was minimized by using a least-squares algorithm where $\mathbf{B}$ and $\mathbf{G}$ were alternatingly updated under non-negativity restric-

tions [133]. More recently, Lee and Seung [113, 114] proposed NMF algorithms which have been used in several machine learning tasks since the algorithms are easy to implement and modify.

Lee and Seung [114] proposed two cost functions and estimation algorithms to obtain $\mathbf{X} \approx \mathbf{BG}$. The cost functions are the square of the Euclidean distance $\mathrm{d}_{\mathrm{euc}}$ and divergence $\mathrm{d}_{\mathrm{div}}$ which are defined as

$$\mathrm{d}_{\mathrm{euc}}(\mathbf{B}, \mathbf{G}) = ||\mathbf{X} - \mathbf{BG}||_F^2 \tag{2.18}$$

and

$$\mathrm{d}_{\mathrm{div}}(\mathbf{B}, \mathbf{G}) = \sum_{k,t} D([\mathbf{X}]_{k,t}, [\mathbf{BG}]_{k,t}), \tag{2.19}$$

where the function D is defined as

$$\mathrm{D}(p, q) = p \log \frac{p}{q} - p + q. \tag{2.20}$$

Both cost functions are lower-bounded by zero, which is obtained only when $\mathbf{X} = \mathbf{BG}$. It can be seen that the Euclidean distance is equal to the first term in (2.15). Minimization of the Euclidean distance leads to a maximum likelihood estimator for $\mathbf{B}$ and $\mathbf{G}$ in the presence of Gaussian noise. Similarly, minimization of the divergence (2.19) leads to a maximum likelihood estimator when each observation $[\mathbf{X}]_{k,t}$ is generated by a Poisson process with mean value $[\mathbf{BG}]_{k,t}$. When $\sum_{k,t}[\mathbf{X}]_{k,t} = \sum_{k,t}[\mathbf{BG}]_{k,t} = 1$, the divergence (2.19) is equal to the Kullback-Leibler divergence, which is widely used as a distance measure between probability distributions [114].

The estimation algorithms of Lee and Seung minimize the chosen cost function by initializing the entries of $\mathbf{B}$ and $\mathbf{G}$ with random positive values, and then by updating them iteratively using multiplicative rules. Each update decreases the value of the cost function until the algorithm converges, i.e., reaches a local minimum. Usually, $\mathbf{B}$ and $\mathbf{G}$ are updated alternatingly.

The update rules for the Euclidean distance are given by

$$\mathbf{B} \leftarrow \mathbf{B}.\times(\mathbf{X}\mathbf{G}^\mathsf{T})./(\mathbf{B}\mathbf{G}\mathbf{G}^\mathsf{T}) \tag{2.21}$$

and

$$\mathbf{G} \leftarrow \mathbf{G}.\times(\mathbf{B}^\mathsf{T}\mathbf{X})./(\mathbf{B}^\mathsf{T}\mathbf{B}\mathbf{G}), \tag{2.22}$$

where $.\times$ and $./$ denote the element-wise multiplication and division, respectively. The update rules for the divergence are given by

$$\mathbf{B} \leftarrow \mathbf{B}.\times\frac{(\mathbf{X}./\mathbf{BG})\mathbf{G}^\mathsf{T}}{\mathbf{1}\mathbf{G}^\mathsf{T}} \tag{2.23}$$

and

$$\mathbf{G} \leftarrow \mathbf{G}.\times\frac{\mathbf{B}^\mathsf{T}(\mathbf{X}./\mathbf{BG})}{\mathbf{B}^\mathsf{T}\mathbf{1}}, \tag{2.24}$$

where $\mathbf{1}$ is an all-one K-by-T matrix, and $\frac{\mathbf{X}}{\mathbf{Y}}$ denotes the element-wise division of matrices $\mathbf{X}$ and $\mathbf{Y}$.

To summarize, the algorithm for NMF is as follows:

(1) Initialize each entry of **B** and **G** with the absolute values of Gaussian noise.

(2) Update **G** using either (2.22) or (2.24) depending on the chosen cost function.

(3) Update **B** using either (2.21) or (2.23) depending on the chosen cost function.

(4) Repeat Steps (2) –(3) until the values converge.

Methods for the estimation of the number of components have not been proposed, but all the methods suggested in Section 2.3 are applicable in NMF, too. The multiplicative update rules have proven to be more efficient than for example the projected steepest-descent algorithms [4, 81, 114].

It has been later noticed (see for example [117]) that minor modifications are needed to guarantee the converge to a stationary point. In our studies we found that neglecting the terms which have zero divisor provides a sufficient convergence, and for simplicity do not present the modifications here.

NMF can be used only for a non-negative observation matrix and therefore it is not suitable for the separation of time-domain signals. However, when used with the magnitude or power spectrogram, the basic NMF can be used to separate components without prior information other than the element-wise non-negativity. In particular, factorization of the magnitude spectrogram using the divergence often produces relatively good results.

NMF does not explicitly aim at components which are statistically independent from each other. However, it has been proved that under certain conditions, the non-negativity restrictions are theoretically sufficient for separating statistically independent sources [138]. It has not been investigated whether musical signals fulfill these conditions, and whether NMF implement a suitable estimation algorithm. Currently, there is no comprehensive theoretical explanation why NMF works so well in sound source separation. If a mixture spectrogram is a sum of sources which have a static spectrum with a time-varying gain, and each of them is active in at least one frame and frequency line in which the other components are inactive, the objective function of NMF is minimized by a decomposition in which the sources are separated perfectly. However, real-world music signals rarely fulfill these conditions. When two or more more sources are present simultaneously at all times, the algorithm is likely to represent then with a single component.

One possibility to motivative NMF is the probabilistic interpretation given by Raj and Smaragdis [143], who considered the gains and basis functions as probability distributions conditional to each component. This allows deriving the multiplicative updates (2.23)-(2.24) from the expectation maximization algorithm [76, pp. 236-242]. Similar motivation for updates was also used by Goto [68, pp. 332-337], who estimated the contribution of pitched sounds of different fundamental frequencies and their spectra in a single frame.

In the analysis of music signals, the basic NMF has been used by Smaragdis and Brown [166], and extended versions of the algorithm have been proposed

for example by Virtanen [189] and Smaragdis [164]. The problem of the large dynamic range of musical signals has been addressed e.g. by Abdallah and Plumbley [4]. By assuming multiplicative gamma-distributed noise in the power spectral domain, they derived the cost function

$$\mathrm{D}(p, q) = \frac{p}{q} - 1 + \log \frac{q}{p} \tag{2.25}$$

to be used instead of (2.20). Compared to the Euclidean distance (2.18) and divergence (2.20), this distance measure is more sensitive to low-energy observations. In our simulations, however, it did not produce as good results as the Euclidean distance or the divergence (see Section 3.3).

2.5 Prior Information about Sources

The above-described algorithms use some general assumptions about the sources in the core algorithms, such as independence or non-negativity, but also other prior information on the sources is often available. For example in the analysis of pitched musical instruments, it is known in advance that the spectra of instruments are approximately harmonic. However, it is difficult to implement harmonicity restrictions in the models discussed earlier.

Prior knowledge can also be source-specific. The most common approach to incorporate prior information about sources in the analysis is to train source-specific basis functions in advance. Several approaches have been proposed. The estimation is usually done in two stages, which are

1. Learn source-specific basis functions from training material, such as monotimbral and monophonic music. Also the characteristics of time-varying gains can be stored, for example by modeling their distribution.

2. Represent a polyphonic signal as a weighted sum of the basis functions of all the instruments. Estimate the gains and keep the basis functions fixed.

Several methods have been proposed for training the basis functions in advance. The most straightforward choice is to separate also the training signal using some of the described methods. For example, Jang and Lee [90] used ISA to train basis functions for two sources separately. Benaroya et al. [15] suggested the use of non-negative sparse coding, but they also tested using the spectra of random frames of the training signal as the basis functions or grouping similar frames to obtain the basis functions. They reported that non-negative sparse coding and the grouping algorithm produced the best results [15]. Non-negative sparse coding was also used by Schmidt and Olsson [155] to both train the basis functions and estimate their gains from a mixture. Gautama and Van Hulle compared three different self-organizing methods in the training of basis functions [62].

The training can be done in a more supervised manner by using a separate set of training samples for each basis function. For example in the drum

transcription systems proposed by FitzGerald et al. [58] and Paulus and Virtanen [136], the basis function for each drum instrument was calculated from isolated samples of each drum. It is also possible to generate the basis functions manually, for example so that each of them corresponds to a single pitch. Lepain used frequency-domain harmonic combs as the basis functions, and parameterized the rough shape of the spectrum using a slope parameter [115]. Sha and Saul trained the basis function for each discrete fundamental frequency using a speech database with annotated pitch [159].

In practice, it is difficult to train basis functions for all the possible sources beforehand. An alternative option is to use trained or generated basis functions which are then adapted to the observed data. For example, Abdallah and Plumbley initialized their non-negative sparse coding algorithm with basis functions that consisted of harmonic spectra with a quarter-tone pitch spacing [4]. After the initialization, the algorithm was allowed to adapt these. For the separation vocals from background music, Ozerov et al. trained a background model using music without vocals [132]. They used an algorithm which segmented an input signal into vocal and non-vocal parts, and then adapted the background model using the non-vocal parts and vocal model using the vocal parts.

Once the basis functions have been trained, the observed input signal is represented using them. Sparse coding and non-negative matrix factorization techniques are feasible also in this task. Usually the reconstruction error between the input signal and the model is minimized while using a small number of active basis functions (sparseness constraint). For example, Benaroya et al. proposed an algorithm which minimizes the energy of the reconstruction error while restricting the gains to be non-negative and sparse [15].

If the sparseness criterion is not used, a matrix $\mathbf{G}$ reaching the global minimum of the reconstruction error can be usually found rather easily. If the gains are allowed to have negative values and the estimation criterion is the energy of the residual, the standard least-squares solution

$$\hat{\mathbf{G}} = (\mathbf{B}^\mathsf{T}\mathbf{B})^{-1}\mathbf{B}^\mathsf{T}\mathbf{X} \tag{2.26}$$

produces the optimal gains (assuming that the priorly-trained basis functions are linearly independent) [97, pp. 220–226]. If the gains are restricted to non-negative values, the least-squares solution is obtained using the non-negative least-squares algorithm [111, p. 161]. When the basis functions, observations, and gains are restricted to non-negative values, the global minimum of the divergence (2.19) between the observations and the model can be computed by applying the multiplicative update (2.24) iteratively [136, 151]. Lepain minimized the sum of the absolute value of the error between the observations and the model by using linear programming and the Simplex algorithm [115].

The estimation of the gains can also be done in a framework which increases the probability of basis functions being non-zero in consecutive frames. For example, Vincent and Rodet used hidden Markov models (HMMs) to model the durations of the tones [184].

It is also possible to train prior distributions for the gains. Jang and Lee used standard ICA techniques to train time-domain basis functions for each source

separately, and modeled the probability distribution function of the component gains with a generalized Gaussian distribution which is a family of density functions of the form $\mathsf{p}(x) \propto \exp(-|x|^q)$ [90]. For an observed mixture signal, the gains were estimated by maximizing their posterior probability.

2.6 Further Processing of the Components

In the processing of music signals the main motivation for separating an input signal into components is that each component usually represents a musically meaningful entity, such as a percussive instrument or all the equal-pitched tones of an instrument. Depending on the application, the separated components can be associated to sources, synthesized, or analyzed to obtain musical information about the sources.

2.6.1 Associating Components with Sources

If the basis functions are estimated from a mixture signal, we do not know which component is produced by which source. Since each source is modeled as a sum of one or more components, we need to associate the components to sources. There are basically two ways to do this. In the unsupervised classification framework, component clusters are formed based on some similarity measure, and these are interpreted as sources. Alternatively, if prior information about the sources is available, the components can be classified to sources based on their distance to source models. Naturally, if pre-trained basis functions are used for each source, the source of each basis function is known and classification is not needed.

Pair-wise dependence between the components can be used as a similarity measure for clustering. Even in the case of ICA which aims at maximizing the independence of the components, some dependencies may remain because it is possible that the input signal contains fewer independent components than are to be separated.

Casey and Westner used the symmetric Kullback-Leibler divergence between the probability distribution functions of basis functions as a distance measure, resulting in an independent component cross-entropy matrix (an "ixegram") [30]. Dubnov proposed a distance measure derived from the higher-order statistics of the basis functions or the gains [45]. Blumensath and Davies [19] used the symmetric Kullback-Leibler divergence between the energies measured within logarithmically spaced frequency bands.

Casey and Westner [30], Dubnov [45], and Blumensath and Davies [19] also suggested clustering algorithms for grouping the components into sources. These try to minimize the inter-cluster dependence and maximize the intra-cluster dependence. However, in our simulations the automatic clustering turned out to be a difficult task (see Section 3.3.3).

For predefined sound sources, the association can be done using pattern recognition methods, which has produced better results. Uhle et al. extracted

acoustic features from each component to classify them either to a drum track or to a harmonic tracks [178]. The features in their system included, for example, the percussiveness of the time-varying gain, and the noise-likeness and dissonance of the spectrum. Another system for separating drums from polyphonic music was proposed by Helén and Virtanen. They trained a support vector machine (SVM) using the components extracted from a set of drum tracks and polyphonic music signals without drums. Different acoustic features were evaluated, including the above-mentioned ones, Mel-frequency cepstral coefficients, and others [79].

2.6.2 Extraction of Musical Information

The separated components can be analyzed to obtain musically important information, such as the onset and offset times and fundamental frequency of each component (assuming that they represent individual tones of a pitched instrument). Naturally, the components can be synthesized and then analyzed using established analysis techniques. However, the synthesis stage is usually not needed, but analysis using the basis functions and gains directly is likely to be more reliable, since the synthesis stage may cause some artifacts.

The onset and offset times of each component j are measured from the time-varying gains $g_{j,t}$, $t = 1 \ldots T$. Ideally, a component is active when its gain is non-zero. In practice, however, the gain may contain interference from other sources and the activity detection has to be done with a more robust method.

Paulus and Virtanen [136] proposed an onset detection procedure that was derived from the psychoacoustically motivated method of Klapuri [99]. The gains of a component were compressed, differentiated, and low-pass filtered. In the resulting "accent curve", all local maxima above a fixed threshold were considered as sound onsets. For percussive sources or other instruments with a strong attack transient, the detection can be done simply by locating local maxima in the gain functions, as done by FitzGerald et al. [58].

The detection of sound offsets is a more difficult problem, since the amplitude envelope of a tone can be exponentially decaying. Methods to be used in the presented framework have not been proposed.

There are several different possibilities for the estimation of the fundamental frequency of a pitched component. For example, prominent peaks can be located from the spectrum and the two-way mismatch procedure of Maher and Beauchamp [118] can be used, or the fundamental period can be estimated from the autocorrelation function which is obtained by inverse Fourier transforming the power spectrum. In our experiments, the enhanced autocorrelation function proposed by Tolonen and Karjalainen [175] was found to produce good results. In practice, a component may represent more than one pitch. This happens especially when the pitches are always present simultaneously, as is the case in a chord for example. No methods have been proposed to detect this situation. Whether a component is pitched or not, can be estimated, e.g., from the acoustic features of the component [79, 178].

Some systems use fixed basis functions which corresponds to certain fundamental frequency values [115, 159]. In this case, the fundamental frequency of each basis function is of course known.

2.6.3 Synthesis

The spectra and gains of the estimated components can be used directly in some acoustic analysis applications. In order to verify the quality by listening, however, the components has to be synthesized to obtain time-domain signals for each source.

Synthesis from time-domain basis functions is straightforward: the signal of component j in frame t is generated by multiplying the basis function $\mathbf{b}_j$ by the corresponding gain $g_{j,t}$, and adjacent frames are combined using the overlap-add method where frames are multiplied by a suitable window function, delayed, and summed.

Synthesis from frequency-domain basis functions is not as trivial. The synthesis procedure includes calculation of the magnitude spectrum of a component in each frame, estimation of the phases to obtain the complex spectrum, and an inverse discrete Fourier transform (IDFT) to obtain the time-domain signal. Adjacent frames are then combined using overlap-add. When magnitude spectra are used as the basis functions, framewise spectra are obtained as the product of the basis function with its gain. If power spectra are used, a square root has to be taken, and if the frequency resolution is not linear, additional processing has to be done to enable synthesis using the IDFT.

A few alternative methods have been proposed for the phase generation. Using the phases of the original mixture spectrogram produces good synthesis quality when the components do not overlap significantly in time and frequency [190]. However, applying the original phases and the IDFT may produce signals which have unrealistic large values at frame boundaries, resulting in perceptually unpleasant discontinuities when the frames are combined using overlap-add. Also the phase generation method proposed by Griffin and Lim [71] has been used in the synthesis (see for example Casey [28]). The method finds phases so that the error between the separated magnitude spectrogram and the magnitude spectrogram of the resynthesized time-domain signal is minimized in the least-squares sense. The method can produce good synthesis quality especially for slowly-varying sources with deterministic phase behavior. The least-squares criterion, however, gives less importance to low-energy partials and often leads to a degraded high-frequency content. Improvements to the above synthesis method have been proposed by Slaney et al. in [162]. The phase generation problem has been recently addressed by Achan et al. who proposed a phase generation method based on a pre-trained autoregressive model [7].

The estimated magnitude spectrograms can be also used to design a Wiener filter, and the sources can be obtained by filtering the mixture signal [14]. The resulting signals resemble those obtained using the original phases; the main difference is that the filtered signals may include parts of the residual of the mixture, which are not included in the separated spectrograms.

Chapter 3

Non-Negative Matrix Factorization with Temporal Continuity and Sparseness Criteria

This chapter proposes an unsupervised sound source separation algorithm, which combines NMF with temporal continuity and sparseness objectives. The algorithm is extended from the work originally published in [189, 191][1]. The additional objectives, especially the temporal continuity, are a simple and efficient way to incorporate knowledge of natural sound sources to the linear signal model. When the additional objectives are introduced, the divergence criterion of NMF may not be minimized with projected steepest descent algorithms. Therefore, an augmented divergence is proposed which can be minimized more robustly. Simulation experiments are carried out to evaluate the performance of the proposed algorithm, which is shown to provide a better separation quality than existing algorithms.

3.1 Signal Model

The algorithm is based on the signal model presented in the previous chapter, where the magnitude spectrum vector $\mathbf{x}_t$ in frame t is modeled as a linear combination of basis functions $\mathbf{b}_j$, which is written as

$$\hat{\mathbf{x}}_t = \sum_{j=1}^{J} g_{j,t}\mathbf{b}_j, \tag{3.1}$$

[1]The text and figures have been adapted from [191] with permission, ©2006 IEEE.

where J is the number of basis functions and $g_{j,t}$ is the gain of the j^{th} basis function in frame t. An observed magnitude spectrogram is modeled as a sum of components $j = 1 \ldots J$, each of which has a fixed spectrum $\mathbf{b}_j$ and a time-varying gain $g_{j,t}$, $t = 1 \ldots T$, T being the number of frames.

The magnitude spectrogram is calculated as follows: first, the time-domain signal is divided into frames and windowed. In our simulations, a fixed 40 ms frame size, Hamming window, and 50% overlap between frames is used. The discrete Fourier transform (DFT) is applied on each frame, the length of the DFT being equal to the frame size. Only positive frequencies are retained and phases are discarded by taking the absolutive values of the DFT spectra, resulting in a magnitude spectrogram matrix $[\mathbf{X}]_{k,t}$ where $k = 1 \ldots K$ is the discrete frequency index and $t = 1 \ldots T$ is the frame index.

As earlier, we write (3.1) using a matrix notation as

$$\hat{\mathbf{X}} = \mathbf{BG}, \tag{3.2}$$

where $\hat{\mathbf{X}} = \left[\hat{\mathbf{x}}_1 \ldots \hat{\mathbf{x}}_T\right]$, $\mathbf{B} = \left[\mathbf{b}_1 \ldots \mathbf{b}_J\right]$ and $[\mathbf{G}]_{j,t} = g_{j,t}$. The observed magnitude spectrogram $\mathbf{X}$ is modeled as a product of the basis matrix $\mathbf{B}$ and the gain matrix $\mathbf{G}$, so that $\mathbf{X} \approx \mathbf{BG}$, while restricting $\mathbf{B}$ and $\mathbf{G}$ to be entry-wise non-negative. This models the linear summation of the magnitude spectrograms of the components. As discussed on Page 18, the summation of power spectra is theoretically better justified; however, the power spectrum representation emphasizes too much high-energy observations. In our simulation experiments (see Section 3.3) the best results were obtained by assuming linear summation of the magnitude spectra and therefore the proposed method is formulated using magnitude spectrograms.

Estimation of $\mathbf{B}$ and $\mathbf{G}$ is done by minimizing a cost function $c(\mathbf{B}, \mathbf{G})$, which is a weighted sum of three terms: a reconstruction error term $c_{\mathrm{r}}(\mathbf{B}, \mathbf{G})$, a temporal continuity term $c_{\mathrm{t}}(\mathbf{G})$, and a sparseness term $c_{\mathrm{s}}(\mathbf{G})$:

$$c(\mathbf{B}, \mathbf{G}) = c_{\mathrm{r}}(\mathbf{B}, \mathbf{G}) + \alpha c_{\mathrm{t}}(\mathbf{G}) + \beta c_{\mathrm{s}}(\mathbf{G}), \tag{3.3}$$

where α and β are the weights for the temporal continuity term and sparseness term, respectively.

3.1.1 Reconstruction Error Function

The divergence cost (2.19) of an individual observation $[\mathbf{X}]_{k,t}$ is linear as a function of the scale of the input, since $D(\gamma p||\gamma q) = \gamma D(p||q)$ for any positive scalar γ, whereas for the Euclidean cost the dependence is quadratic. Therefore, the divergence is more sensitive to small-energy observations, which makes it more suitable for the estimation of perceptually meaningful sound sources (see Page 18). Among the tested reconstruction error measures (including also those proposed in [4] and [190]), the divergence produced the best results, and therefore we measure the reconstruction error using the divergence.

The divergence (2.19) approaches infinity as the value of the model $[\mathbf{BG}]_{k,t}$ approaches zero. The spectrograms of natural audio signals have a large dynamic range, and small values of $[\mathbf{X}]_{k,t}$ and $[\mathbf{BG}]_{k,t}$ are therefore probable. The

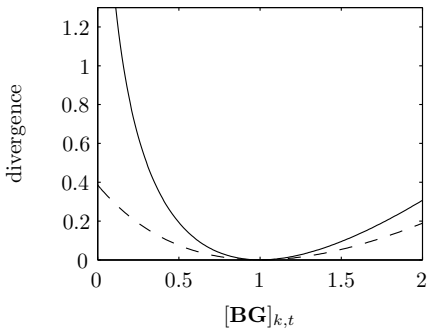

Figure 3.1: Illustration of the divergence (solid line) and augmented divergence (dashed line), with values $[\mathbf{X}]_{k,t} = 1$ and $\epsilon = 1$. The normal divergence approaches infinity as $[\mathbf{BG}]_{k,t}$ approaches zero, while the augmented divergence is finite at $[\mathbf{BG}]_{k,t} = 0$.

parameters $\mathbf{B}$ and $\mathbf{G}$ are estimated by iterative methods, which alternatively update the gains and the spectra. The projected gradient descent algorithm updates the parameters into the direction of the negative gradient, and then projects the parameters to non-negative values. Because of the shape of the divergence function, the convergence of the projected steepest descent is often poor, and it tends to stuck easily into a local minimum.

To improve the convergence of the optimization algorithm, we propose an augmented divergence function D_ϵ, defined as

$$D_\epsilon(\mathbf{X}||\mathbf{BG}) = \sum_{k,t}([\mathbf{X}]_{k,t} + \epsilon) \log \frac{[\mathbf{X}]_{k,t} + \epsilon}{[\mathbf{BG}]_{k,t} + \epsilon} - [\mathbf{X}]_{k,t} + [\mathbf{BG}]_{k,t} \tag{3.4}$$

in which the parameter ϵ is positive. The augmented divergence is finite at $[\mathbf{BG}]_{k,t} = 0$, as seen in Fig. 3.1. Similarly to the normal divergence, the augmented divergence between non-negative matrices is always non-negative, and zero only if $\mathbf{X} = \mathbf{BG}$.

Different ways of determining the value ϵ were tested. Setting it equal to $\epsilon = \frac{1}{TK}\sum_{t=1}^{T}\sum_{k=1}^{K}[\mathbf{X}]_{k,t}$, which is the average of all the bins of the magnitude spectrogram, was found to produce good results, when the performance was measured using the simulations presented in Section 3.3. The performance with different values of ϵ is evaluated in Section 3.3.

3.1.2 Temporal Continuity Criterion

The separation methods discussed in Chapter 2 considered each frame as an individual observation. However, real-world sounds usually have a temporal structure, and their acoustic characteristics vary slowly as a function of time. Fig. 3.2 shows a simple example where the temporal continuity criterion would

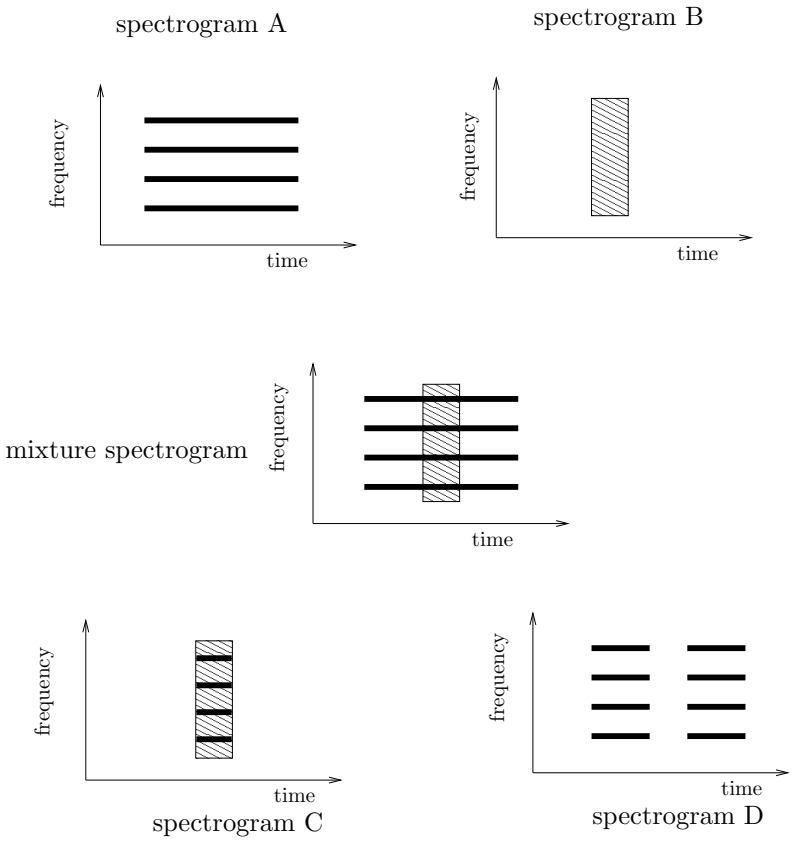

Figure 3.2: A simple example which illustrates how the temporal continuity of sources can improve the separation. See text for details.

increase the robustness of the separation. The two sources A and B represent a typical sustained harmonic sound and a typical short-duration percussive sound, respectively. The observed mixture spectrogram (illustrated in the middle panel) is separated into two components both of which have a fixed spectrum and time-varying gain. When the separation is done by minimizing the reconstruction error between the observed spectrogram and the separated spectrograms, it is possible to obtain the original spectrograms A and B. However, it is also possible to represent the mixture spectrogram as a sum of spectrograms C and D, resulting in error. By favoring temporal continuity, the separation can be directed towards the spectrograms A and B.

Temporal continuity was aimed at in the system proposed Vincent and Rodet who modeled the activity of a source by a hidden Markov model [184]. In this chapter, we apply a simple temporal continuity objective which does not require training beforehand.

Temporal continuity of the components is measured by assigning a cost to

large changes between the gains $g_{j,t}$ and $g_{j,t-1}$ in adjacent frames. We propose to use the sum of the squared differences between the gains. To prevent the numerical scale of the gains from affecting the cost, the gains are normalized by their standard deviation estimates σ_j, so that the cost function c_t for the temporal continuity can be written as:

$$c_\mathrm{t}(\mathbf{G}) = \sum_{j=1}^{J} \frac{1}{\sigma_j^2} \sum_{t=2}^{T} (g_{t,j} - g_{t-1,j})^2 \tag{3.5}$$

If the normalization was not used, the function $c_\mathrm{t}(\mathbf{G})$ could be minimized without affecting the reconstruction error by scaling the matrices by $\mathbf{B} \leftarrow \mathbf{B}\theta$ and $\mathbf{G} \leftarrow \mathbf{G}/\theta$, where θ is a large positive scalar. The standard deviation of each component $j = 1 \ldots J$ is estimated as

$$\sigma_j = \sqrt{\frac{1}{T} \sum_{t=1}^{T} g_{t,j}^2}. \tag{3.6}$$

In [189], a cost function was used which was the sum of the absolute values of the difference of gains of adjacent frames. The motivation for using the absolute value cost was that, e.g., for a gain rising from a level to another, the cost is equal for all monotonically increasing transitions. However, it was found that the absolute value of the differences did not increase the performance of the separation algorithm as much as the squared differences. The reason for this might be the steepest descent optimization algorithm: the gradient of the temporal continuity cost depends only on the sign of the difference, not on the magnitude.

3.1.3 Sparseness Objective

The sparseness of the activities $g_{j,t}$ has been utilized in several blind source separation algorithms, and there are cases where the sparseness criterion improves the quality. For example, when the spectrum of one source (e.g. kick drum) covers partly the spectrum of another (e.g. snare drum), the latter source could be modeled as a sum of the first sound and a residual. The use of sparse gains can favor a representation where only a single spectrum is used to model the latter source.

The sparseness objective, which is derived from the MAP estimation of the sources can be formulated as as a sum of function f taken of all the elements of $\mathbf{G}$ (see Section 2.3). To prevent the scale of the gains from affecting the objective, we normalize the gains of a component by its standard deviation, to result in sparseness objective

$$c_\mathrm{s}(\mathbf{G}) = \sum_{j=1}^{J} \sum_{t=1}^{T} f(g_{j,t}/\sigma_j), \tag{3.7}$$

where $f(\cdot)$ is a function which penalizes non-zero gains. We used $f(x) = |x|$, which has been used, for example, by Hoyer [81] in the separation of synthesized images and by Benaroya et al. [15] in the separation of priorly trained audio signals. The method allows also other choices for f.

3.2 Estimation Algorithm

In the estimation algorithm, the matrices $\mathbf{B}$ and $\mathbf{G}$ are first initialized with random positive values and then alternatively updated. The value of the cost function decreases at each update, until the algorithm converges. A multiplicative update rule is used to update $\mathbf{B}$, and either a multiplicative update rule or projected steepest descent is used to update $\mathbf{G}$.

Currently there is no reliable method for the automatic estimation of the number of components, but it has to be set manually. In practice, a large number of components can be used, which are then clustered to sound sources. If there is some prior information of the sources, it can be used to select the number of components or to initialize the spectra.

The source matrix $\mathbf{G}$ is updated either using a multiplicative update rule or the projected steepest descent method. Both methods require estimating the gradient of total cost $c(\mathbf{B}, \mathbf{G})$, which is the weighted sum of the gradients of the reconstruction error, temporal error, and sparseness error, given by

$$\nabla c(\mathbf{B}, \mathbf{G}) = \nabla c_{\mathrm{r}}(\mathbf{B}, \mathbf{G}) + \alpha \nabla c_{\mathrm{t}}(\mathbf{G}) + \beta \nabla c_{\mathrm{s}}(\mathbf{G}) \tag{3.8}$$

The gradients of the terms in (3.8) with respect to $\mathbf{G}$ are given by

$$\nabla c_{\mathrm{r}}(\mathbf{B}, \mathbf{G}) = \mathbf{B}^{\mathsf{T}}(\mathbf{1} - \frac{\mathbf{X} + \epsilon}{\mathbf{B}\mathbf{G} + \epsilon}), \tag{3.9}$$

where $\mathbf{1}$ is a all-one matrix of the same size as $\mathbf{X}$,

$$\begin{aligned}[\nabla c_{\mathrm{t}}(\mathbf{G})]_{j,t} =& 2T \frac{2g_{j,t} - g_{j,t-1} - g_{j,t+1}}{\sum_{u=1}^{T} g_{j,u}^2} \\ & - \frac{2T g_{j,t} \sum_{u=2}^{T} (g_{j,u} - g_{j,u-1})^2}{(\sum_{u=1}^{T} g_{j,u}^2)^2},\end{aligned} \tag{3.10}$$

and

$$[\nabla c_{\mathrm{s}}(\mathbf{G})]_{j,t} = \frac{1}{\sqrt{\frac{1}{T}\sum_{u=1}^{T} g_{j,u}^2}} - \frac{\sqrt{T} g_{j,t} \sum_{u=1}^{T} g_{j,u}}{(\sum_{u=1}^{T} g_{j,u}^2)^{3/2}}, \tag{3.11}$$

respectively.

A multiplicative update is derived as in [151] by first writing the total gradient (3.8) as a subtraction $\nabla c(\mathbf{B}, \mathbf{G}) = \nabla c^{+}(\mathbf{B}, \mathbf{G}) - \nabla c^{-}(\mathbf{B}, \mathbf{G})$ of element-wise non-negative terms

$$\nabla_c^{+}(\mathbf{B}, \mathbf{G}) = \nabla c_{\mathrm{r}}^{+}(\mathbf{B}, \mathbf{G}) + \alpha \nabla c_{\mathrm{t}}^{+}(\mathbf{G}) + \beta \nabla c_{\mathrm{s}}^{+}(\mathbf{G}) \tag{3.12}$$

and

$$\nabla_c^-(\mathbf{B},\mathbf{G}) = \nabla c_{\mathrm{r}}^-(\mathbf{B},\mathbf{G}) + \alpha \nabla c_{\mathrm{t}}^-(\mathbf{G}) + \beta \nabla c_{\mathrm{s}}^-(\mathbf{G}), \tag{3.13}$$

where the element-wise positive terms of the gradients of reconstruction error cost, temporal continuity cost, and sparseness cost are given by

$$\nabla c_{\mathrm{r}}^+(\mathbf{B},\mathbf{G}) = \mathbf{B}^{\mathsf{T}}\mathbf{1}, \tag{3.14}$$

$$\nabla c_{\mathrm{r}}^-(\mathbf{B},\mathbf{G}) = \mathbf{B}^{\mathsf{T}}\frac{\mathbf{X}+\epsilon}{\mathbf{B}\mathbf{G}+\epsilon}, \tag{3.15}$$

$$[\nabla c_{\mathrm{t}}^+(\mathbf{G})]_{j,t} = \frac{4T g_{j,t}}{\sum_{u=1}^T g_{j,u}^2}, \tag{3.16}$$

$$[\nabla c_{\mathrm{t}}^-(\mathbf{G})]_{j,t} = 2T\frac{g_{j,t-1}+g_{j,t+1}}{\sum_{u=1}^T g_{j,u}^2} + \frac{2T g_{j,t}\sum_{u=2}^T (g_{j,u}-g_{j,u-1})^2}{(\sum_{u=1}^T g_{j,u}^2)^2}, \tag{3.17}$$

$$[\nabla c_{\mathrm{s}}^+(\mathbf{G})]_{j,t} = \frac{1}{\sqrt{\frac{1}{T}\sum_{u=1}^T g_{j,u}^2}}, \tag{3.18}$$

and

$$[\nabla c_{\mathrm{s}}^-(\mathbf{G})]_{j,t} = \frac{g_{j,t}\sqrt{T}\sum_{u=1}^T g_{j,u}}{(\sum_{u=1}^T g_{j,u}^2)^{3/2}}. \tag{3.19}$$

The terms (3.14)-(3.19) are element-wise non-negative, since the gains, basis functions, and observations are restricted to non-negative values.

Multiplicative update for $\mathbf{G}$ is then given as

$$\mathbf{G} \leftarrow \mathbf{G}.\times\frac{\nabla c^-(\mathbf{B},\mathbf{G})}{\nabla c^+(\mathbf{B},\mathbf{G})}. \tag{3.20}$$

In the cost function (3.3), $\mathbf{B}$ affects only the reconstruction error term $c_{\mathrm{r}}(\mathbf{B},\mathbf{G})$, so that the gradient of the total cost with respect to $\mathbf{B}$ is

$$\nabla c(\mathbf{B},\mathbf{G}) = (\mathbf{1} - \frac{\mathbf{X}+\epsilon}{\mathbf{B}\mathbf{G}+\epsilon})\mathbf{G}^{\mathsf{T}}. \tag{3.21}$$

Factoring the gradient into positive and negative terms as in (3.14) and (3.15) results in a multiplicative update rule

$$\mathbf{B} \leftarrow \mathbf{B}.\times\frac{\frac{\mathbf{X}+\epsilon}{\mathbf{B}\mathbf{G}+\epsilon}\mathbf{G}^{\mathsf{T}}}{\mathbf{1}\mathbf{G}^{\mathsf{T}}}. \tag{3.22}$$

An alternative gain estimation technique is the projected steepest descent algorithm, which updates the parameters into the direction of the negative gradient, and then projects them into non-negative values. It has been previously used to estimate non-negative sparse representation, for example, by Hoyer [82] and Virtanen [189].

The overall iterative algorithm is the following:

1. Initialize each element of $\mathbf{B}$ and $\mathbf{G}$ with the absolute value of Gaussian noise.

2. Update $\mathbf{B}$ using the multiplicative update rule (3.22).

3.(a) Update the gains using (3.20), or

3.(b) Update the gains using projected steepest descent:

- Calculate the gradient of $c(\mathbf{B}, \mathbf{G})$ with respect to $\mathbf{G}$ using (3.8).
- Update $\mathbf{G} \leftarrow \mathbf{G} - \mu \nabla c(\mathbf{B}, \mathbf{G})$. The positive step size μ is adaptively varied using the bold driver algorithm [150] explained below.
- Set negative entries of $\mathbf{G}$ to zero.

4. Evaluate the value of the cost function $c(\mathbf{B}, \mathbf{G})$.

The steps 2...4 are repeated until the value of the cost function converges. The iteration is stopped when the decrease has been smaller than a predefined threshold for a certain number of iterations. For a 10-second input signal and 20 components the algorithm takes a couple of hundred iterations to converge, a couple of minutes of computation on a 1.7 GHz desktop PC when implemented in Matlab.

In the projected steepest descent, the bold driver algorithm updates the step size as follows. After iterations where the value of the cost function decreases, the step size is increased by a small factor (by 5%). When the value of the cost function increases, the step size is reduced rapidly (by 50%). This procedure was found to provide faster convergence than a fixed step size. In our simulations it was also found advantageous to use the moment term [150] of the gradient, so that the effective gradient used in the updating is the weighted sum of gradients of current and previous iteration.

The multiplicative update rules are good in the sense in practice that they do not require additional parameters like the steepest descent algorithms, where a suitable step size has to be estimated. The multiplicative updates guarantee the non-negativity of the parameters, since both the numerator and denominator are always element-wise non-negative. Similarly to gradient algorithms, the multiplicative update rules will not change the parameter values when a stationary point is reached, since the multiplier becomes unity at a stationary point.

When the temporal continuity and sparseness term are not used ($\alpha = 0$ and $\beta = 0$), the cost function (3.3) can be shown to be always non-increasing under the proposed multiplicative update rule (3.22), as shown in Appendix A.1. Since $D_\epsilon(\mathbf{X}||\mathbf{BG}) = D_\epsilon(\mathbf{X}^\mathsf{T}||\mathbf{G}^\mathsf{T}\mathbf{B}^\mathsf{T})$, the roles of $\mathbf{G}$ and $\mathbf{B}$ can be swapped to change (3.22) into (3.20), when $\alpha = 0$ and $\beta = 0$. This shows that the cost function (3.3) is non-increasing also under the update rule (3.20), when $\alpha = 0$ and $\beta = 0$. In the case of the multiplicative updates the use of augmented divergence is not necessary, so that $\epsilon = 0$ can also be used.

When $\alpha > 0$ or $\beta > 0$, the multiplicative update (3.20) does not necessarily decrease the value of the cost function. In the simulation experiments presented

in Section 3.3, we applied the multiplicative update rules on 300 signals, each of which was tested with 4 different component counts and several combinations of α and β (see Fig 3.3). We observed a total of 5 cases where the value of the cost function increased, which took places when α had a large value. Minimizing the cost function by projected steepest descent led to almost identical results, with the expense of increased computational complexity. This and the small amount of cost increases show that also the multiplicative updates are sufficient for minimizing the cost function.

3.3 Simulation Experiments

3.3.1 Acoustic Material

Test signals were generated by mixing samples of pitched musical instruments and drums. The pitched sounds were from a database of individual tones which is a combination of samples from the McGill University Master Samples Collection [131], the University of Iowa website [89], and samples recorded from the Roland XP-30 synthesizer. The instruments introduce several sound production mechanisms, variety of spectra and also modulations, such as vibrato. The total number of samples available for generating the mixtures was 4128, each having the sampling frequency 44100 Hz. The drum samples were from the DFH Superior commercial sample database [44], which contains individual drum hits from several drum kits and instruments. Each instrument in the DFT Superior database is multi-sampled, i.e., the recording is repeated several times for each instrument.

Mixture signals were generated by choosing a random number of pitched instrument sources and a random number of drum sources. For each mixture, the number of sources was chosen randomly from within the limits shown in Table 3.1. Once the number of sources had been chosen, each source was picked randomly from the databases. For pitched-instrument sources, a random instrument and a random fundamental frequency from the available samples were allotted and for drum sources, a random drum kit and a random drum instrument were allotted.

Each pitched instrument sample was used only once within a mixture, and they were truncated to random lengths. We used a random number of repetitions of each drum tone, which were unique samples. The location of each note was randomized by allotting a random onset time between 0 and 6 seconds. The length of all source signals was chosen to be 7 seconds. This resulted in material where 79% of the frames contained more than one source, i.e., the sources here mainly overlapping.

To simulate the mixing conditions encountered in real-world situations, each source was scaled to obtain a random total energy between 0 and -20 dB. The reference source signals before mixing were stored to allow the measurement of the separation quality. The total number of mixtures was 300. Examples of the mixture signals are available for listening at `http://www.cs.tut.fi/~tuomasv/`.

Table 3.1: Parameters used to generate the test signals.

parameter	interval
number of pitched instrument sources	[0 12]
number of drum sources	[0 6]
length of each pitched musical sound (s)	[0.15 1]
number of notes per each drum source	[2 8]
onset time of each repetition (s)	[0 6]
energy of each source (dB)	[0 -20]

It should be noted that the acoustic material differs from real-world music in a sense that it consists of individual notes instead of note sequences. However, none of the tested methods is able to utilize different pitch values of a source in the separation, and it is therefore unlikely that the results would be significantly different if note sequences were present.

3.3.2 Tested Algorithms

Some recently published algorithms were used as a baseline in the evaluation. All the algorithms apply a 40 ms window length, Hanning window, and short-time Fourier transform to calculate a spectrogram $\mathbf{X}$ of the mixture signal, as described in the beginning of Section 3.1. Unless otherwise mentioned, the methods operate on the magnitude spectrogram. The following algorithms were tested:

- Independent subspace analysis (ISA). Implementation of the algorithm follows the outline proposed by Casey and Westner [30], but instead of aiming at statistically independent gains we estimated statistically independent spectra, since that produced better results. Independent components were estimated using the FastICA algorithm [54, 84].
- NMF was tested with the algorithms proposed in [114]. These minimize the divergence or the Euclidean distance, and are denoted by NMF-DIV and NMF-EUC, respectively.
- The non-negative sparse coding algorithm proposed by Abdallah & Plumbley [4] is denoted by NMF-LOG, since the method roughly minimizes the distance between the logarithm of the spectra. It assumes that the sources sum in the power spectral domain, so that the observation vector and basis functions in (3.1) are power spectra.

The proposed algorithm was evaluated by using different configurations. The weights α and β were not optimized for the test data, but different magnitudes (1,10,100...) were tried using similar test cases, and the values $\alpha = 100$ and $\beta = 0$ which approximately produced the best results were chosen. The effect of the weights is illustrated in the next section. Both the multiplicative (denoted

by proposed-NMF$_{\text{multi}}$) and projected steepest descent (denoted by proposed-NMF$_{\text{proj}}$) estimation algorithms were tested. For the multiplicative algorithm the augmented term in the divergence is not required, and therefore it was tested with $\epsilon = 0$.

3.3.3 Evaluation Procedure

Each mixture signal was separated into components using all the algorithms. Since there is no reliable method for the estimation of the number of the components, all the methods were tested with 5, 10, 15, and 20 components, and the results were averaged.

Since the tested methods are unsupervised, we do not know which component belongs to which source, and supervised clustering cannot be used. On the other hand, manual clustering is too troublesome and unreliable. This enforces us to automatic clustering, which requires that the original signals before mixing are used as references for clusters. The signal-to-distortion ratio between the separated and and an original signal is an obvious choice for determining the source of the separated signal. To prevent the synthesis procedure from affecting the quality, the measure was calculated between the magnitude spectrograms $\mathbf{Y}_m$ and $\hat{\mathbf{Y}}_j$ of the m^{th} reference and j^{th} separated component, respectively:

$$\text{SDR}(m, j) = \frac{\sum_{k,t} [\mathbf{Y}_m]_{k,t}^2}{\sum_{k,t} \left([\mathbf{Y}_m]_{k,t} - [\hat{\mathbf{Y}}_j]_{k,t}\right)^2}. \tag{3.23}$$

A component j is assigned to a source m which produces the maximum SDR.

A large number of components which are clustered using the original signals as references may produce unrealistically good results, since in practice there does not exist a clustering algorithm which could produce as good separated signals. We tested the unsupervised clustering methods proposed in [30] and [45], trying to create component clusters for each source. However, these deteriorated the results in all the cases.

To overcome these problems we modeled each source with a single component for which the SDR was largest. This approach utilizes a minimum amount of prior information about the reference signals, but still produces applicable results.

The quality of the separated sources was also measured by calculating the signal-to-distortion ratio (SDR) between the original magnitude spectrogram $\mathbf{Y}$ and corresponding separated magnitude spectrogram $\hat{\mathbf{Y}}$, given in dB by

$$\text{SDR[dB]} = 10 \log_{10} \frac{\sum_{k,t} [\mathbf{Y}]_{k,t}^2}{\sum_{k,t} \left([\mathbf{Y}]_{k,t} - [\hat{\mathbf{Y}}]_{k,t}\right)^2}. \tag{3.24}$$

The SDR (in dB) was averaged over all the sources and mixtures to get the total measure of the separation performance. If no components were assigned to a source, the source was defined to be undetected. The detection error rate

Table 3.2: Simulation results of the unsupervised learning algorithms based on the linear model. The best result in each column is highlighted in bold.

algorithm	detection error rate (%)			SDR (dB)		
	all	pitched	drums	all	pitched	drums
ISA	35	34	35	4.6	5.4	2.8
NMF-EUC	28	28	30	6.6	7.9	**3.7**
NMF-DIV	26	28	23	7.0	8.8	3.5
NMF-LOG	80	90	57	2.3	2.7	2.2
proposed-NMF_{proj}	**23**	**24**	23	**7.3**	**9.1**	3.6
proposed-$\text{NMF}_{\text{multi}}$	24	25	**22**	**7.3**	**9.1**	3.6

was defined as the ratio of the total number of undetected sources and the total number of sources. The undetected sources were not used in the calculation of the average SDR.

3.3.4 Results

The average SDRs and detection rates are shown in Table 3.2. The averages are shown for all sources and for pitched and drum sounds separately. The 95% confidence intervals [80, pp. 212-219] for the average detection rate and the SDR were smaller than $\pm 1\%$ and ± 0.1dB, respectively, for all the algorithms.

Use of the temporal continuity term improves the detection of pitched sounds significantly. The proposed method also enables a slightly better SDR of pitched sources than NMF-DIV. If also undetected sources were included in the computing the SDR, the improvement would be larger. The differences between the multiplicative and projected steepest descent algorithms is small.

The non-negative matrix factorization algorithms NMF-EUC and NMF-DIV produce clearly better results than ISA, and the overall performance of NMF-DIV is better than NMF-EUC. For drum sources the improvement gained by the proposed methods are small, which is natural since drum sounds are temporally less continuous than pitched instruments.

The performance of NMF-LOG is poor according to the detection error rate and SDR. These measures are derived from the energy of the error signal, but NMF-LOG is much more sensitive to low-energy observations. To investigate this further, we used also the likelihood proposed in [4] to measure the quality of the sources. The likelihood is based on a multiplicative noise model, and it results in a distortion measure $\sum_{k,t}([\mathbf{Y}]_{k,t}^2/[\hat{\mathbf{Y}}]_{k,t}^2 - 1 + \log([\hat{\mathbf{Y}}]_{k,t}^2/[\mathbf{Y}]_{k,t}^2))$ between the original magnitude spectrogram $\mathbf{Y}$ and the separated magnitude spectrogram $\hat{\mathbf{Y}}$ (a small positive constant was added to both terms to avoid numerical problems in the divisions). The measure is more sensitive to low-energy observations than the SDR. The distortion measures were 0.30 (ISA), 2.38 (NMF-EUC), 1.94 (NMF-DIV), 5.08 (NMF-LOG), 1.02 (proposed-$\text{NMF}_{\text{multi}}$), and 1.02 (proposed-$\text{NMF}_{\text{proj}}$), a smaller value indicating a better quality. This shows that the chosen performance measure has some effect on the results, since accord-

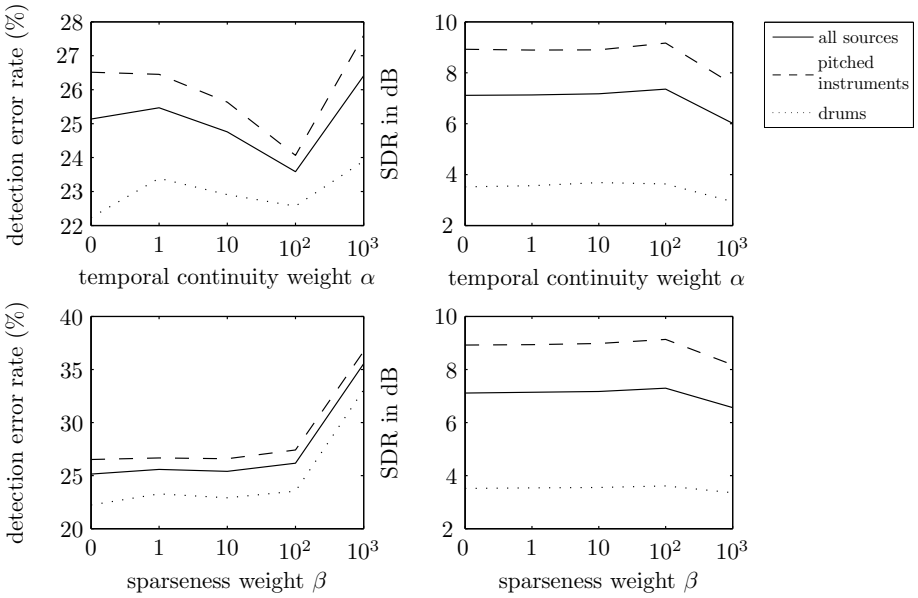

Figure 3.3: The effect of different temporal continuity weights α and sparseness weights β on the detection error rate and signal-to-distortion ratio of proposed-NMF$_{\text{proj}}$, when the other parameter was 0.

ing to this measure ISA gives the best quality, although the order of the other methods remains the same. Unlike ISA, the NMF-based algorithms do not allow subtraction of components, and therefore often produce values $[\hat{\mathbf{Y}}]_{k,t} \approx 0$, which results in a large distortion measure. The quality analysis in this paper is mainly based on SDR, since it is more widely used.

The effect of the weights α and β is illustrated in Fig. 3.3. The use of the temporal continuity term ($\alpha > 0$) improves especially the detection of pitched instrument sources. The sparseness term ($\beta > 0$) was not found to improve the results. When either value is too large, the quality of the separation degrades clearly.

The effect of the augmented divergence and the optimization algorithm was evaluated by testing the proposed method with the multiplicative update rules and the projected steepest descent, while not using the temporal continuity or sparseness terms (by setting $\alpha = 0$ and $\beta = 0$). The performance of the algorithms with different values of ϵ is illustrated in Fig. 3.4. Also the normal divergence was tested. With suitable values of ϵ (scaling factor 1 or 10) the performance of the augmented divergence is marginally better than with the normal divergence. When ϵ gets smaller, the performance of the steepest descent algorithm decreases significantly, while the performance of the multiplicative update algorithm approaches the performance of the normal divergence. This illustrates that the projected steepest descent algorithm is not feasible in the

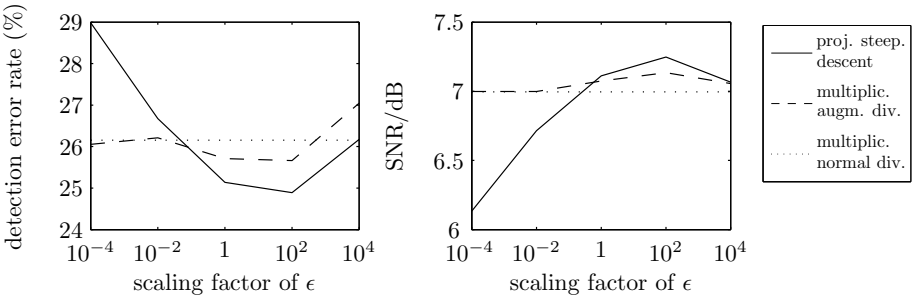

Figure 3.4: Illustration of the effect of the optimization algorithm and term ϵ on the augmented divergence. The solid line is the proposed method estimated with the projected steepest descent algorithm and the dashed line is the proposed method estimated with the multiplicative update rules with weights $\alpha = 0$ and $\beta = 0$ with different values of ϵ (the scaling factor times the average of $\mathbf{X}$). The dotted line is the normal divergence, for reference. reference.

minimization of the normal divergence ($\epsilon = 0$).

With scaling factor 1 the multiplicative update rules resulted in a 1.1% lower average value of the cost function than the steepest descent algorithm. This verifies the earlier implications that from optimization point of view, the multiplicative updates are more efficient in NMF tasks. However, lower average value of the cost function did not result to better detection error rate or SDR, as can be seen in Figure 3.4.

In our implementation the iteration was stopped when the ratio of the cost function values between two consecutive iterations was smaller than a predefined threshold $(1 + 10^{-5})$ for a certain number of iterations. The steepest descent algorithm required approximately twice the number of iterations compared to the multiplicative updates, which is natural since an optimal step size parameter in the steepest descent algorithm was not estimated at each iteration.

Figure 3.5 illustrates the performance of proposed-NMF$_{\text{proj}}$, NMF-DIV, and ISA as a function of the number of components, separately for the cases where either a single component or all the components were clustered using the original signals as references. The latter case was included because the original idea of ISA is based on the use of multiple components per source. The detection error rate of both algorithms approaches zero as the number of components increases. The proposed method enables a better detection error rate with all component counts. Increasing the number of components increases the average SDR of the separated sources up to a certain point, after which it decreases in the cases for single-component algorithms, and saturates for multiple-component algorithms. When all the components are used, the asymptotic SDR of NMF-DIV is better than proposed-NMF$_{\text{proj}}$. However, the SDR of both algorithms is limited, which suggests that non-negativity restrictions alone are not sufficient for high-quality separation, but further assumptions such as harmonicity of the sources or a more flexible signal model might be needed.

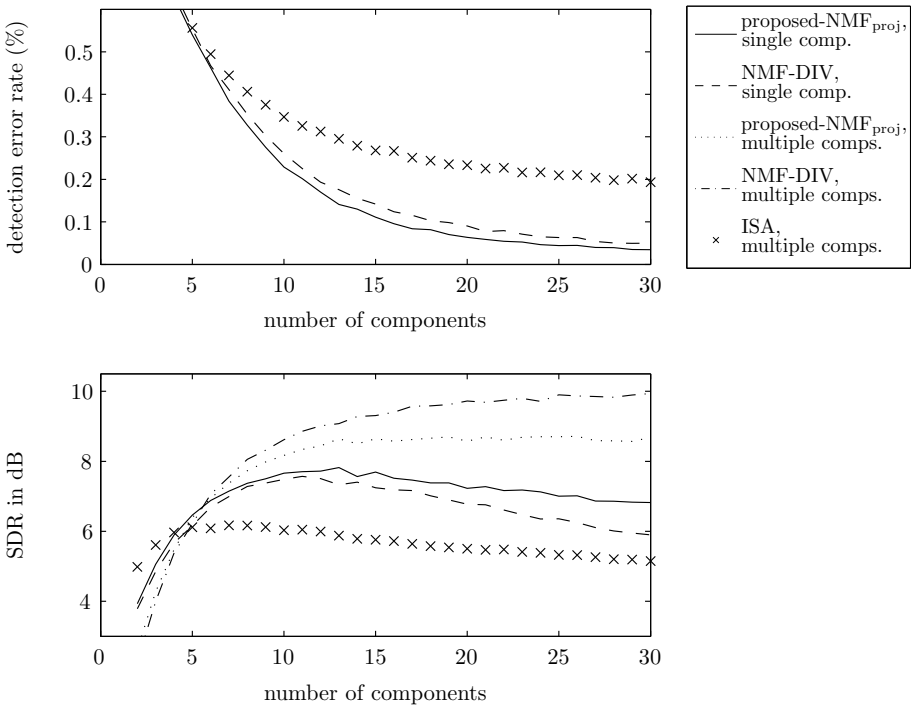

Figure 3.5: Illustration of the effect of the component count. 'Single comp.' refers to measures where a single component was used to model each source, and 'multiple comps.' refers to measures where all the components were clustered using the original signals are references. The detection error rate is not affected by the use of multiple components per source, and hence only three lines can be seen in the upper panel.

Chapter 4

Time-Varying Components

The linear instantaneous model (2.1) discussed in two preceding chapters is efficient in the analysis of music signals since many musically meaningful entities can be rather well approximated with a fixed spectrum and a time-varying gain. However, sources with a strongly time-varying spectrum have to be represented as a sum of multiple components. Furthermore, each fundamental frequency value produced by a pitched instrument has to be represented with a different component.

Increasing the number of components makes their estimation more difficult, and this also requires a clustering algorithm which associates components to their sources. As discussed in Section 2.6.1, the clustering is a difficult task at least when the sources are not known beforehand. Instead of using multiple components per source, more complex models can be used which allow either a time-varying spectrum or a time-varying fundamental frequency for each component. These two models are discussed separately in Sections 4.1 and 4.2. The first model was originally published in [190].

4.1 Time-Varying Spectra

The signal model presented in this section is based on the assumption that a sound source generates events so that the spectrogram of each occurrence is similar. The representation is obtained by extending the model (2.1) so that a single basis function $\mathbf{b}_j$ is replaced by L basis functions $\mathbf{b}_{j,\tau}$, $\tau = 0 \ldots L-1$. In consecutive frames they assemble an *event spectrogram*, where τ is the frame index of the event, and L the duration of the event in frames.

In the signal model (2.1), multiplication is replaced by convolution, resulting in the model

$$\mathbf{x}_t = \sum_{j=1}^{J} \sum_{\tau=0}^{L-1} \mathbf{b}_{j,\tau} g_{j,t-\tau}. \tag{4.1}$$

This can be given the following interpretation: the event spectrogram $\mathbf{b}_{j,\tau}$ mod-

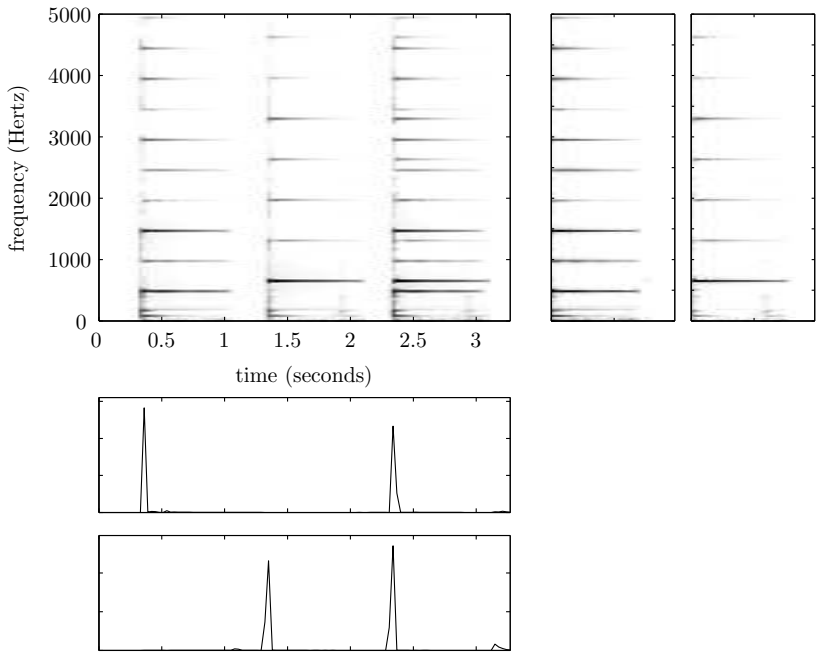

Figure 4.1: An example of the convolutive model (4.1) which allows time-varying components. The mixture spectrogram (upper left panel) contains the tones C#6 and F#6 of the acoustic guitar, first played separately and then together. The upper right panels illustrate the learned tone spectrograms and the lower panels show their time-varying gains. In the gains, an impulse corresponds to the onset of a tone. The components were estimated by minimizing the divergence using the multiplicative updates presented in this chapter.

els the spectrogram of one event of a component j, and non-zero values of $g_{j,t}$ describe the locations in which the events of component j set on. The values of the non-zero gains also describe the scale of each repetition. A simple two-tone example is illustrated in Figure 4.1.

Since the model (4.1) is a convolution between the event spectrogram and its time-varying gain, it is called a *convolutive signal model*, and later the term *non-negative matrix deconvolution* (NMD) is used to refer to the estimation algorithm which is extended from NMF. In the analysis of audio signals the convolutive signal model has previously been used in the NMF framework by Smaragdis [164] and in non-negative sparse coding framework by Virtanen [190]. Time-varying components have also been used in vision research to analyze time-varying sparse representations [20], [128], [129]. Also the method proposed by Blumensath and Davies [18] can be formulated using (4.1). Their objective was to find sparse and shift-invariant decompositions of a signal in the time domain. Also in the multichannel separation framework the instantaneous linear model

has been extended to model convolutive mixtures [177]. In these, the effect of the transfer path from each source to each sensor is modeled by convolution with the impulse response of the transfer path.

The convolutive signal model suits well for representing various musical sound sources. Basically it enables learning any repeating structures from an input signal. The repetition of a source is a requirement for its separation; if an event occurs only once, it is unlikely to become separated from the other co-occurring sounds. Also, if two or more sound sources are always present simultaneously, the model is likely to represent them with a single component.

By modeling sound sources as a sum of repeating events, we can utilize the temporal correlation present in natural sound sources. As discussed in Section 3.1.2, many estimation algorithms based on the linear model (2.1) consider each frame as an individual observation. The temporal continuity term in Section 3 aimed at slowly-varying gains of components, but the convolutive model is able to adapt to any repeating spectral structure.

The convolutive signal model is particularly suitable for percussive instruments, the repetitions of which are approximately similar. An example of a drum loop is illustrated in Figure 4.2. Each impulse in the estimated time-varying gains corresponds to an onset of a drum hit. Ideally the convolutive model produces impulses at the locations of the onsets of events, while the time-varying gains produced by the linear model (2.1) follow the amplitude or energy envelope of a source. For more complex noisy signals it is difficult to obtain exact impulses, but still the learned components correspond to individual sources, and can be used in their separation.

As Figure 4.1 illustrates, also harmonic sounds can be represented using the model. If a note of a pitched musical instrument is played similarly several times, its spectrogram can be learned from the mixture signal. However, pitched instrument tones usually have a longer duration, and arbitrarily long durations L cannot be used if the basis functions are estimated from a mixture signal. When $JL \geq T$, the input spectrogram can be represented perfectly as a sum of concatenated event spectrograms (without separation). Meaningful sources are likely to be separated only when $JL \ll T$. In other words, estimation of several components with large L requires long input signals. With long input signals and event spectrogram durations the proposed algorithm becomes computationally slow, and therefore the separation of individual tones of pitched instruments from polyphonic signals could not be tested systematically. Naturally, the model can also represent sustained harmonic sounds with approximately fixed spectrum, and also ones with constant-rate vibrato; one period of the vibrato is represented using the event spectrogram.

By using a slightly modified version of the the notation as suggested by Smaragdis in [164, 165], the model (4.1) can be written in a matrix form as

$$\hat{\mathbf{X}} = \sum_{\tau=0}^{L-1} \mathbf{B}_\tau \cdot \underset{\tau\rightarrow}{\mathbf{G}} \tag{4.2}$$

where $\mathbf{B}_\tau = \left[\mathbf{b}_1, \ldots, \mathbf{b}_J\right]$, $[\mathbf{G}]_{j,t} = g_{j,t}$, and $\underset{\tau\rightarrow}{\mathbf{G}}$ is a shift operator, which moves

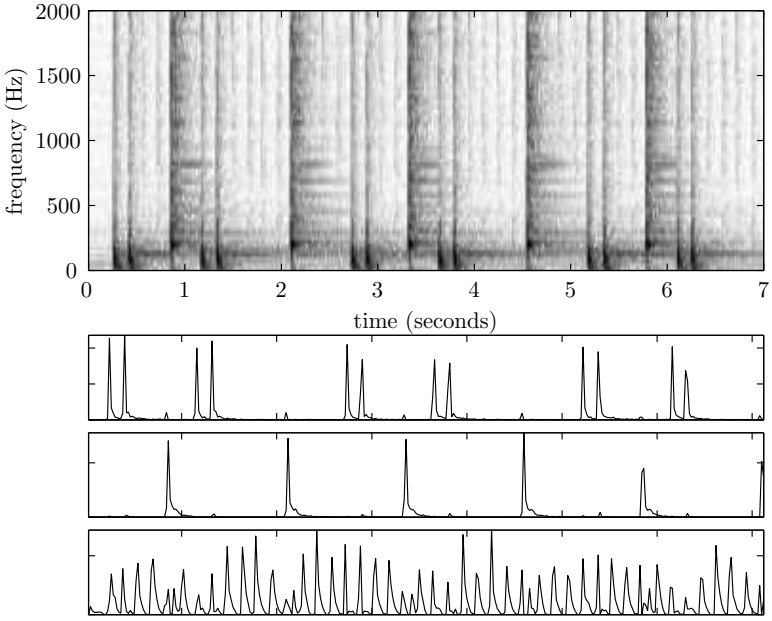

Figure 4.2: An example of the convolutive model for percussive sources. A spectrogram of a drum pattern consisting of bass drum, snare drum, and hi-hats is illustrated in the upper plot. The gains of three components separated from the signal are illustrate in the lower plots. The onset of each drum event corresponds to an impulse in the gains.

the columns of $\mathbf{G}$ by τ indices to the right. The shifting does not change the size of the matrix, but τ rightmost columns are discarded and τ columns are zero-padded to the left. Formally, the shifting can be defined as

$$[\underset{\tau\rightarrow}{\mathbf{G}}]_{j,t} = \begin{cases} [\mathbf{G}]_{j,t-\tau} & t > \tau \\ 0 & t \leq \tau \end{cases} \tag{4.3}$$

Similarly, a shift operator to the left is defined as

$$[\underset{\leftarrow\tau}{\mathbf{G}}]_{j,t} = \begin{cases} [\mathbf{G}]_{j,t+\tau} & t \leq T - \tau \\ 0 & t > T - \tau \end{cases} \tag{4.4}$$

where T is the number of columns in $\mathbf{G}$.

4.1.1 Estimation Algorithms

The matrices $\mathbf{G}_\tau$ and $\mathbf{B}$ in the convolutive model (4.2) can be estimated using methods extended from NMF and sparse coding. In these, the reconstruction error between the model and the observations is minimized, while restricting $\mathbf{G}_\tau$ and $\mathbf{B}$ to be entry-wise non-negative. Favoring sparse gains can potentially improve results, since real-world sound events set on in a small number of frames only. In [164] Smaragdis proposed an algorithm which aims at minimizing the divergence between the observation and the model while constraining non-negativity, whereas in [190] we proposed an algorithm which is based on non-negative sparse coding. Also Blumensath and Davies [18] included a sparse prior for the gains. Their model uses temporal resolution of one sample, which makes the dimensionality of the optimization problem large.

The estimation algorithms presented here require that an input signal is represented using either the magnitude or power spectrogram, so that non-negativity restrictions can be used. We obtained good results using the magnitude spectrogram calculated using a fixed 40 ms window length and the DFT. As earlier, the spectrum vector in frame $t = 1 \ldots T$ is denoted by $\mathbf{x}_t$, and the matrix $\mathbf{X} = \begin{bmatrix} \mathbf{x}_1 \ldots \mathbf{x}_T \end{bmatrix}$ is used to denote the observed magnitude spectrogram.

Feasible measures for the reconstruction error c_r between the observations $\mathbf{X}$ and the model $\hat{\mathbf{X}}$ are at least the earlier presented Euclidean distance (2.18), divergence (2.19), and augmented divergence (3.4). When sparseness objective is included, the total cost function c is a sum of reconstruction error term c_r and sparseness term c_s:

$$c(\mathbf{G}, \mathbf{B}_0, \ldots, \mathbf{B}_{L-1}) = c_\mathrm{r}(\mathbf{G}, \mathbf{B}_0, \ldots, \mathbf{B}_{L-1}) + \beta c_\mathrm{s}(\mathbf{G}), \tag{4.5}$$

where β is the weight of the sparseness cost. The sparseness term can be defined, for example, as in (3.7).

The design of multiplicative update rules for the minimization of the objective requires calculating the gradient of the objective. The gradient of the

Euclidean distance $\mathrm{d}_{\mathrm{euc}}$ with respect to $\mathbf{G}$ is given by

$$\nabla_{\mathbf{G}} \mathrm{d}_{\mathrm{euc}}(\mathbf{G}, \mathbf{B}_0, \ldots, \mathbf{B}_{L-1}) = \sum_{\tau=0}^{L-1} \mathbf{B}_\tau^{\mathsf{T}} \cdot (\underset{\tau\leftarrow}{\mathbf{X}} - \underset{\tau\leftarrow}{\hat{\mathbf{X}}}) \tag{4.6}$$

and gradient of the divergence $\mathrm{d}_{\mathrm{div}}$ with respect to $\mathbf{G}$ by

$$\nabla_{\mathbf{G}} \mathrm{d}_{\mathrm{div}}(\mathbf{G}, \mathbf{B}_0, \ldots, \mathbf{B}_{L-1}) = \sum_{\tau=0}^{L-1} \mathbf{B}_\tau^{\mathsf{T}} \cdot [\underset{\tau\leftarrow}{\mathbf{1}} - (\frac{\underset{\tau\leftarrow}{\mathbf{X}}}{\underset{\tau\leftarrow}{\hat{\mathbf{X}}}})], \tag{4.7}$$

where $\mathbf{1}$ is a all-one matrix of the same size as $\mathbf{X}$.

The gradients of the Euclidean distance and the divergence with respect to $\mathbf{B}_\tau$ are given by

$$\nabla_{\mathbf{B}_\tau} \mathrm{d}_{\mathrm{euc}}(\mathbf{G}, \mathbf{B}_0, \ldots, \mathbf{B}_{L-1}) = (\mathbf{X} - \hat{\mathbf{X}}) \cdot \underset{\tau\rightarrow}{\mathbf{G}}^{\mathsf{T}} \tag{4.8}$$

and

$$\nabla_{\mathbf{B}_\tau} \mathrm{d}_{\mathrm{div}}(\mathbf{G}, \mathbf{B}_0, \ldots, \mathbf{B}_{L-1}) = (\mathbf{1} - \frac{\mathbf{X}}{\hat{\mathbf{X}}}) \cdot \underset{\tau\rightarrow}{\mathbf{G}}^{\mathsf{T}}, \tag{4.9}$$

respectively. The gradient of the sparseness cost (3.7) was given in (3.11).

As in Section 3.2, the terms are factored into the sum of positive and negative terms. Then, the update rules for the minimization of the weighted sum of the Euclidean distance and the sparseness cost are given by

$$\mathbf{G} \leftarrow \mathbf{G}. \times \frac{\left(\sum_{\tau=0}^{L-1} \mathbf{B}_\tau^{\mathsf{T}} \cdot \underset{\tau\leftarrow}{\mathbf{X}}\right) + \beta \nabla c_{\mathrm{s}}^{-}(\mathbf{G})}{\left(\sum_{\tau=0}^{L-1} \mathbf{B}_\tau^{\mathsf{T}} \cdot \underset{\tau\leftarrow}{\hat{\mathbf{X}}}\right) + \beta \nabla c_{\mathrm{s}}^{-}(\mathbf{G})} \tag{4.10}$$

and

$$\mathbf{B}_\tau \leftarrow \mathbf{B}_\tau. \times \frac{\mathbf{X} \cdot \underset{\tau\rightarrow}{\mathbf{G}}^{\mathsf{T}} + \beta \nabla c_{\mathrm{s}}^{-}(\mathbf{G})}{\hat{\mathbf{X}} \cdot \underset{\tau\rightarrow}{\mathbf{G}}^{\mathsf{T}} + \beta \nabla c_{\mathrm{s}}^{-}(\mathbf{G})}, \tag{4.11}$$

where $\hat{\mathbf{X}}$ is obtained from (4.2),and the positive and negative gradients $\nabla c_{\mathrm{s}}^{+}$ and $\nabla c_{\mathrm{s}}^{-}$ of the sparseness cost (3.7) were given in (3.18) and (3.19)

The update rules for the minimization of the weighted sum of the divergence and the sparseness cost are given by

$$\mathbf{G} \leftarrow \mathbf{G}. \times \frac{\left[\sum_{\tau=0}^{L-1} \mathbf{B}_\tau^{\mathsf{T}} \cdot (\frac{\underset{\tau\leftarrow}{\mathbf{X}}}{\underset{\tau\leftarrow}{\hat{\mathbf{X}}}})\right] + \beta \nabla c_{\mathrm{s}}^{-}(\mathbf{G})}{\left(\sum_{\tau=0}^{L-1} \mathbf{B}_\tau^{\mathsf{T}} \cdot \underset{\tau\leftarrow}{\mathbf{1}}\right) + \beta \nabla c_{\mathrm{s}}^{+}(\mathbf{G})} \tag{4.12}$$

and

$$\mathbf{B}_\tau \leftarrow \mathbf{B}_\tau. \times \frac{(\frac{\mathbf{X}}{\hat{\mathbf{X}}}) \cdot \underset{\tau\rightarrow}{\mathbf{G}}^{\mathsf{T}} + \beta \nabla c_{\mathrm{s}}^{-}(\mathbf{G})}{\mathbf{1} \cdot \underset{\tau\rightarrow}{\mathbf{G}}^{\mathsf{T}} + \beta \nabla c_{\mathrm{s}}^{+}(\mathbf{G})}, \tag{4.13}$$

When $\beta = 0$, the update (4.13) reduces to the one proposed by Smaragdis in [164], but the rule (4.12) is slightly different. Unlike the update rule for $\mathbf{G}$ presented in [164], (4.12) can be shown to converge. The convergence proofs for the update rules are given in the Appendix A.2. The update rules for the augmented divergence (3.4) can be obtained from (4.12) and (4.13) by replacing $[\frac{\mathbf{X}}{\mathbf{X}}]$ by $[\frac{\mathbf{X}+\epsilon}{\mathbf{X}+\epsilon}]$.

The overall estimation algorithm is similar to the algorithm presented on page 40: the matrices $\mathbf{B}$ and $\mathbf{G}_\tau$ are initialized with random positive values, and they are then updated alternatively until the values converge. Both the number of components and L, the duration of event spectrograms have to be set manually. In our simulations, event spectrogram durations between 1 and 25, and number of components between 1 and 35 were tested.

Synthesis In the synthesis, the spectrum of component $j = 1 \dots J$ within each frame $t = 1 \dots T$ is calculated as $\sum_{\tau=0}^{L-1} g_{j,t-\tau} \mathbf{b}_{j,\tau}$. A time-domain signal can be obtained from this by applying the methods described in 2.6.3.

4.1.2 Simulation Experiments

Simulation experiments were conducted using the same material and test procedure as in Sec. 3.3. The non-negative matrix deconvolution algorithms which minimize the Euclidean distance and divergence without the sparseness term ($\beta = 0$) using the update rules (4.10)-(4.13), are denoted by NMD-EUC and NMD-DIV, respectively. The case $\beta > 0$ is examined separately. The algorithm proposed in [164] is denoted by NMD-DIV(*).

Because the computational complexity of these algorithms is larger than those discussed in Chapter 3, we had to use a slightly more sensitive stopping criterion. Therefore, the results presented in Section 3.3 are not directly comparable to the ones presented here. However, we tested here the NMD algorithms also with $L = 1$, which correspond to basic NMF algorithms, and are here denoted by NMF-EUC and NMF-DIV, respectively.

In the first experiment we used 10 components, which is approximately equal to the average number of sources within the mixtures, and $L = 5$, which was found to lead to good results on the average. The average SDRs and detection error rates are shown on Table 4.1. The averages are calculated for all sources, and also separately for pitched-instrument and drum sources. When the reconstruction error measures are examined separately, the NMD algorithms give better results than the NMF algorithms, except in the detection of pitched instruments, where NMF-DIV performs better than NMD-DIV. The performance of NMD-DIV is approximately equal to the performance of NMD-DIV(*).

In the second experiment, we tested different values for L while the number of components was fixed to 10. The performance of NMD-EUC and NMD-DIV as a function of L is illustrated in Figure 4.3. When L is increased slightly (by 1) from 1, both the detection and SDR improve. Larger L gives diverse results: it increases the detection error rate of pitched instruments, which indicates that

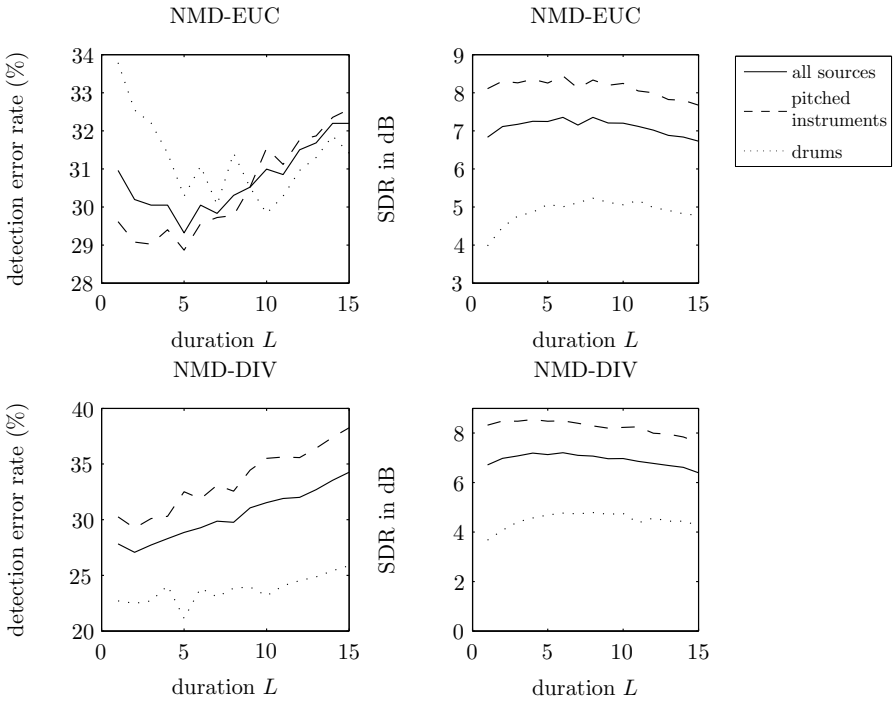

Figure 4.3: The average SDR and detection error rate as a function of the event spectrogram duration L for NMD-EUC (upper plots) and NMD-DIV (lower plots). The solid line is the average of all sources, the dashed line is the average of pitched instrument sources, and the dotted line is the average of drum sources.

Table 4.1: Simulation results of non-negative matrix deconvolution algorithms. The best result in each column is highlighted in bold.

algorithm	SDR (dB)			detection error rate (%)		
	all	pitched	drums	all	pitched	drums
NMD-EUC	**7.2**	8.3	**5.1**	29	**29**	30
NMD-DIV	7.1	**8.5**	4.7	29	33	**21**
NMD-DIV(*)	7.0	8.3	4.6	**28**	30	23
NMF-EUC	6.8	8.1	4.0	31	30	34
NMF-DIV	6.7	8.3	3.7	**28**	30	23

the algorithm does not help in learning individual pitched tones. For NMD-EUC, increasing L improves the detection of drums significantly, and it also improves their average SDR. Larger L also increases the SDR of drum sources for NMD-DIV. For pitched instruments the effect of SDR is smaller, even though small improvements can be obtained. The best average results were obtained using L between 5 and 10, which correspond to spectrogram lengths between 100 and 200 ms, respectively. The lengths are sufficient for representing most drum hits, and the performance measures show that the model is suitable for representing drum signals.

In the third experiment, the number of components was varied while keeping $L = 5$ fixed. The average SDR and detection error rate as a function of the number of components is illustrated in Figure 4.4. Increasing the number of components naturally increases the detection rate monotonically. However, larger number of components does not lead to significantly better SDRs: in the case of pitched sources the quality saturates after approximately 10 components, and in the case of drum components after 3 components.

The effect of the sparseness term was tested by using different values of β together with the divergence. The results are illustrated in Figure 4.5. The use of sparseness term ($\beta > 0$) does not significantly improve the average results, and large values of β lead to a clearly degraded performance.

4.2 Time-Varying Fundamental Frequencies

Compared with the earlier presented models which represent each fundamental frequency value with a different component, a model which can represent different fundamental frequencies of an instrument with a single component provides the advantage that the spectrum of a tone of a certain fundamental frequency can be predicted using tone of adjacent fundamental frequency values. In the case where the tones of an instrument are present with co-occurring sources, estimating their spectrum and separating them individually is difficult. However, when tones with adjacent fundamental frequency values are estimated jointly, the shape of the spectrum can often be deduced, and separation becomes more reliable. Furthermore, this approach produces representations which are a good

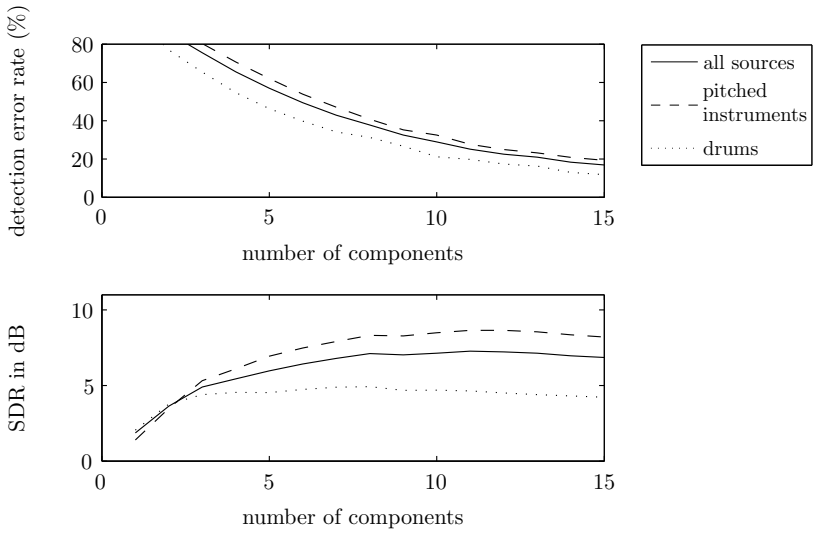

Figure 4.4: The average SDR and detection error rate of NMD-DIV as a function of the number of components, when the duration L was 5. The solid line is the average of all sources, the dashed line is the average of pitched instrument sources, and the dotted line is the average of drum sources.

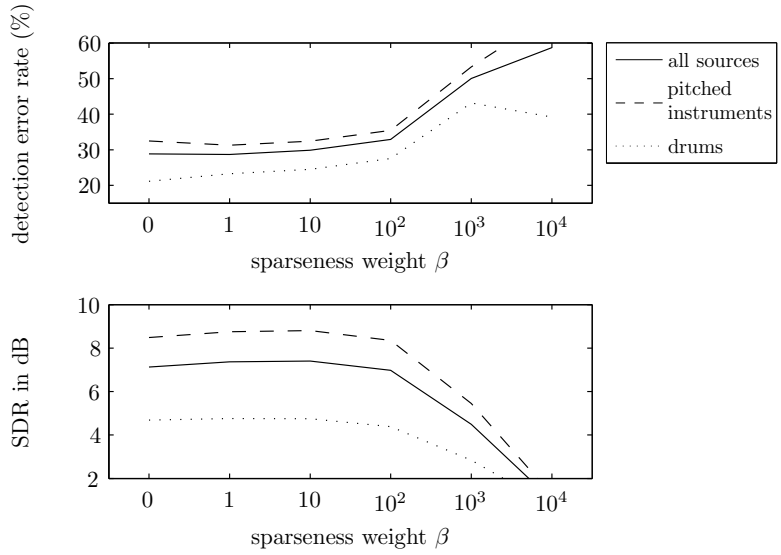

Figure 4.5: The effect of different sparseness weights β on the detection error rate and SDR. The solid line is the average of all sources, the dashed line is the average of pitched sounds, and the dotted line is the average of drums.

and intuitive basis for automatic transcription of pitched instruments.

Varying fundamental frequencies are difficult to model using time-domain basis functions or frequency-domain basis functions with linear frequency resolution. This is because changing the fundamental frequency of a basis function is a non-linear operation which is difficult to implement with these representations: if the fundamental frequency is multiplied by a factor γ, the frequencies of the harmonic components are also multiplied by γ; this can be viewed as a stretching of the spectrum. For an arbitrary value of γ, the stretching is difficult to perform on a discrete linear frequency resolution, at least using a simple operator which could be used for non-parametric representations. The same holds as well for time-domain basis functions.

A logarithmic spacing of frequencies bins makes it easier to represent varying fundamental frequencies. A logarithmic scale consists of discrete frequencies $\mathrm{f_{ref}}\zeta^{k-1}$, where $k = 1 \ldots K$ is the discrete frequency index, $\zeta > 1$ is the ratio between adjacent frequency bins, and $\mathrm{f_{ref}}$ is a reference frequency in Hertz which can be selected arbitrarily. For example, $\zeta = \sqrt[12]{2}$ produces a frequency scale where the spacing between the frequencies is one semitone. In practice, a power spectrum of a logarithmic frequency resolution can be calculated by applying the FFT to obtain linear-resolution power spectrum, and then calculating the energy within each log-frequency band by summing power spectrum bins weighted by the frequency response of the log-frequency band [24].

On the logarithmic scale, the spacing of the partials of a harmonic sound is independent of its fundamental frequency. For fundamental frequency f_0, the overtone frequencies of a perfectly harmonic sound are are mf_0, where m is a positive integer. On the logarithmic scale, the corresponding frequency indices are $k = \log_\zeta(m) + \log_\zeta(f_0/\mathrm{f_{ref}})$, and thus the fundamental frequency affects only the offset $\log_\zeta(f_0/\mathrm{f_{ref}})$, not the intervals between the harmonics.

Given the spectrum $X(k)$ of a harmonic sound with fundamental frequency f_0, a fundamental frequency multiplication γf_0 can be implemented simply as a translation $\hat{X}(k) = X(k - \delta)$, where δ is given by $\delta = \log_\zeta \gamma$. Compared with the stretching of the spectrum, this is much easier to implement.

The estimation of harmonic spectra and their translations can be done adaptively by fitting a model onto the observations.[1] However, this is difficult for an unknown number of sounds and fundamental frequencies, since the reconstruction error as a function of translation δ has several local minima at harmonic intervals, which makes the optimization procedure likely to stuck into a local minimum far from the global optimum. A more feasible parameterization allows each component to have several active fundamental frequencies in each frame, the amount of which is to be estimated. This means that each time-varying gain $g_{j,t}$ is replaced by gains $g_{j,t,z}$, where $z = 0 \ldots Z$ is a fundamental frequency shift index and Z is the maximum allowed shift. The gain $g_{j,t,z}$ describes the amount of the j^{th} component in frame t at a fundamental frequency which is obtained by translating the fundamental frequency of basis function $\mathbf{b}_j$ by z indices.

[1] This approach is related to the fundamental frequency estimation method of Brown, who calculated the cross-correlation between an input spectrum and a single harmonic template on the logarithmic frequency scale [23].

The size of the shift z depends on the frequency resolution. For example, if 48 frequency lines within each octave are used ($\zeta = \sqrt[48]{2}$), $z = 4$ corresponds to a shift of one semitone. For simplicity, the model is formulated to allow shifts only to higher frequencies, but it can be formulated to allow both negative and positive shifts, too.

A vector $\mathbf{g}_{j,t} = \left[g_{j,t,0}, \ldots, g_{j,t,Z}\right]^\mathsf{T}$ is used to denote the gains of component j in frame t. The model can be formulated as

$$\hat{\mathbf{x}}_t = \sum_{j=1}^{J} \mathbf{b}_j * \mathbf{g}_{j,t}, \qquad t = 1 \ldots T, \tag{4.14}$$

where $*$ denotes a convolution operator, defined between vectors as

$$\mathbf{y} = \mathbf{b}_j * \mathbf{g}_{j,t} \quad \Leftrightarrow \quad y_k = \sum_{z=0}^{Z} b_{j,k-z} g_{j,t,z}, \quad k = 1 \ldots K. \tag{4.15}$$

Fig. 4.6 shows the basis function and gains estimated from the example signal in Fig. 2.1.

4.2.1 Estimation Algorithm

In general, the parameters can be estimated by fitting the model to observations with certain restrictions, such as non-negativity or sparseness. Algorithms for this purpose can be derived by extending those used in NMF and sparse coding. Here we present an extension of NMF, where the parameters are estimated by minimizing either the Euclidean distance or the divergence (2.19) between the observations $\mathbf{X}$ and the model (4.14), while restricting the gains and basis functions to be non-negative. Again, the elements of $\mathbf{g}_{j,t}$ and $\mathbf{b}_j$ are initialized with random values and then updated iteratively until the values converge.

The update rule of gains for the minimization of the Euclidean distance is given by

$$\mathbf{g}_{j,t} \leftarrow \mathbf{g}_{j,t} . \times \frac{\mathbf{b}_j \star \mathbf{x}_t}{\mathbf{b}_j \star \hat{\mathbf{x}}_t}, \qquad j = 1, \ldots, J, \qquad t = 1, \ldots, T \tag{4.16}$$

where $\star$ denotes the correlation of vectors, defined for real-valued vectors $\mathbf{b}_j$ and $\mathbf{x}_t$ by

$$\mathbf{y} = \mathbf{b}_j \star \mathbf{x}_t \quad \Leftrightarrow \quad y_k = \sum_{z=0}^{Z} b_{j,k} x_{t,k+z}, \quad k = 1 \ldots K. \tag{4.17}$$

The update rule for the basis functions is given by

$$\mathbf{b}_j \leftarrow \mathbf{b}_j . \times \frac{\sum_{t=1}^{T} \mathbf{g}_{j,t} \star \mathbf{x}_t}{\sum_{t=1}^{T} \mathbf{g}_{j,t} \star \hat{\mathbf{x}}}, j = 1, \ldots, J, \tag{4.18}$$

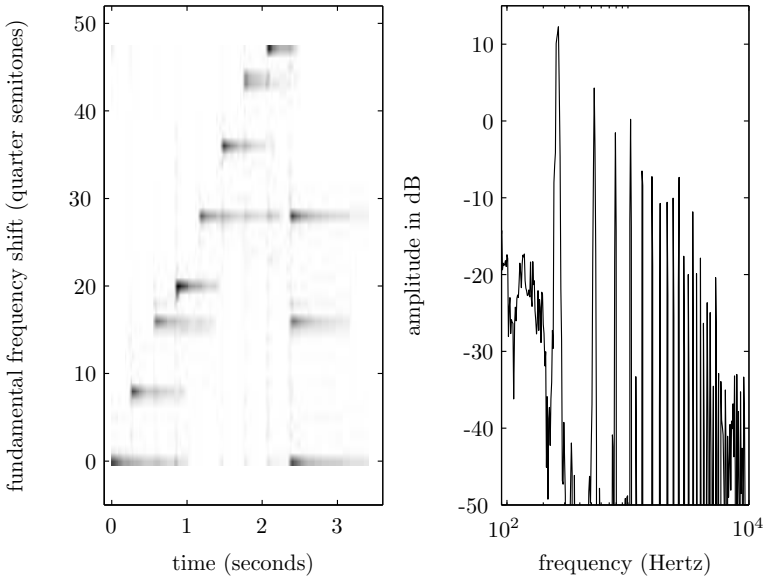

Figure 4.6: Illustration of the time-varying gains (left) and the basis function (right) of a component that was estimated from the example signal in Fig. 2.1 containing a diatonic scale and C major chord. On the left, the intensity of the image represents the value of the gain at each fundamental frequency shift and frame index. Here the fundamental frequencies of the tones can be seen more clearly than from the spectrogram of Fig. 2.1. The parameters were estimated using the algorithm proposed in this chapter which minimizes the divergence.

where the correlations are calculated for delays $1 \dots K$, to result in a vector of length K.

Similarly, the update rules for the minimization of the divergence are given as

$$\mathbf{g}_{j,t} \leftarrow \mathbf{g}_{j,t} . \times \frac{\mathbf{b}_j \star (\frac{\mathbf{x}_t}{\tilde{\mathbf{x}}_t})}{\mathbf{b}_j \star \mathbf{1}}, \qquad j = 1, \dots, J, \qquad t = 1, \dots, T, \tag{4.19}$$

and

$$\mathbf{b}_j \leftarrow \mathbf{b}_j . \times \frac{\sum_{t=1}^{T} \mathbf{g}_{j,t} \star (\frac{\mathbf{x}_t}{\tilde{\mathbf{x}}_t})}{\sum_{t=1}^{T} \mathbf{g}_{j,t} \star \mathbf{1}}, \qquad j = 1, \dots, J, , \tag{4.20}$$

where $\mathbf{1}$ is a K-length vector of ones.

Note that it is possible to constrain the basis functions to a harmonic spectrum of a predefined fundamental frequency by initializing parts of the spectrum to zero values, in which case the multiplicative updates cannot change them.

The algorithm produces good results if the number of sources is small, but for multiple sources and more complex signals, it is difficult to get as good results as those illustrated in Fig. 4.6. The model allows all the fundamental frequencies within the range $z = 0 \dots Z$ to be active simultaneously, which can produce undesirable results: for example, the algorithm may model a non-harmonic drum spectrum by using a harmonic basis function shifted to multiple adjacent fundamental frequencies. Ideally, this could be solved by restricting the gains to be sparse, but the sparseness criterion complicates the optimization.

When shifting the harmonic structure of the spectrum, the formant structure becomes shifted, too. Therefore, representing time-varying pitch by translating the basis function is appropriate only for nearby pitch values. It is unlikely that the whole fundamental frequency range of an instrument could be modeled by shifting a single basis function. This can be resolved either by using different components for different fundamental frequency regions, or in a single-source case by whitening the observed spectrum using an inverse linear prediction filter.

4.3 Dualism of the Time-Varying Models

The models presented in Sections 4.1 and 4.2 are each others duals: by changing the representation, the update rules and estimation algorithms become exactly the same. Therefore, it is sufficient to implement only one of the algorithms, and then change the representation to get the other. Here we shortly describe how parameters of the time-varying frequency model can be estimated using the estimation algorithm for time-varying spectra, and vice versa.

Estimating time-varying fundamental frequencies using the algorithm for time-varying spectra First, the original mixture spectrogram $\tilde{\mathbf{X}}$ is calculated. As explained on Page 59, logarithmic frequency resolution has to be used. The input observation matrix $\mathbf{X}$ for the estimation algorithm is the transpose $\mathbf{X} = \tilde{\mathbf{X}}^{\mathsf{T}}$ of the spectrogram. Now we can apply the update rules presented in

Section 4.1. The event spectrogram length L should be set equal to the desired maximum pitch shift Z plus one.

The outputs of the algorithm are the gain matrix $\mathbf{G}$ and spectra $\mathbf{B}_\tau$, $\tau = 0, \ldots, L-1$. The parameters of the time-varying frequency model can be obtained from these by

$$\tilde{b}_{j,\tilde{k}} = [\mathbf{G}]_{j,t}, \qquad \tilde{k} = t = 1, \ldots, T \tag{4.21}$$

and

$$\tilde{g}_{j,\tilde{t},z} = [\mathbf{B}_\tau]_{k,j}, \qquad \tilde{t} = k = 1, \ldots, K, \qquad z = \tau = 0, \ldots, Z \tag{4.22}$$

Thus, the frame index t in the time-varying spectrum model corresponds to the frequency index $\tilde{k}$ in the time-varying frequency model and vice versa, and the time shift index τ corresponds to the pitch shift index z.

Estimating time-varying spectra Time-varying spectra can be estimated using the algorithm for time-varying frequencies by using the same procedure as above: first, the original mixture spectrogram $\tilde{\mathbf{X}}$ is calculated. The input observation matrix $\mathbf{X}$ for the estimation algorithm is the transpose $\mathbf{X} = \tilde{\mathbf{X}}^\mathsf{T}$ of the spectrogram, and the parameters are estimated using update rules presented in Section 4.2. The maximum pitch shift Z is set equal to the event spectrogram length L minus one.

The outputs of the algorithm are the gains $g_{j,t,z}$ and basis functions $\mathbf{b}_j$. The parameters of the time-varying spectrum model can be obtained from these by

$$[\tilde{\mathbf{G}}]_{j,\tilde{t}} = b_{j,k}, \qquad \tilde{t} = k = 1, \ldots, K \tag{4.23}$$

and

$$[\tilde{\mathbf{B}}_\tau]_{\tilde{k},j} = g_{j,t,z}, \qquad \tilde{k} = t = 1, \ldots, T, \qquad z = \tau = 0, \ldots, Z \tag{4.24}$$

4.4 Combining the Time-Varying Models

It is possible to include components with time-varying spectra and time-varying fundamental frequencies into a same model. Simply put, the combined model $\hat{\mathbf{X}}^{\text{tot}}$ the sum of the model $\hat{\mathbf{X}}^{\text{spec}}$ (4.1) for time-varying spectra and the model $\hat{\mathbf{X}}^{\text{freq}}$ (4.14) for time-varying fundamental frequencies:

$$\hat{\mathbf{X}}^{\text{tot}} = \hat{\mathbf{X}}^{\text{spec}} + \hat{\mathbf{X}}^{\text{freq}} \tag{4.25}$$

In the following, we use the superscripts $^{\text{spec}}$ and $^{\text{freq}}$ to distinguish between the matrices $\mathbf{G}^{\text{spec}}$ and $\mathbf{B}_\tau^{\text{spec}}$ of the time-varying spectrum model and the vectors $\mathbf{g}_{j,t}^{\text{freq}}$ and $\mathbf{b}_j^{\text{freq}}$ of the time-varying fundamental frequency model.

The estimation can done using earlier presented principles, i.e., by minimizing the reconstruction error while keeping the parameters non-negative. This can be carried out by applying update rules presented earlier. One possibility for this is the following algorithm:

(1) Initialize each entry of the matrices $\mathbf{G}^{\text{spec}}$ and $\mathbf{B}_{\tau}^{\text{spec}}$ and vectors $\mathbf{g}_{j,t}^{\text{freq}}$ and $\mathbf{b}_{j}^{\text{freq}}$ with the absolute values of Gaussian noise.

(2) Evaluate the combined model $\hat{\mathbf{X}}^{\text{tot}}$ (4.25).

(3) Update $\mathbf{G}^{\text{spec}}$ using either (4.10) or (4.12) depending on the chosen cost function, while replacing the model $\hat{\mathbf{X}}$ in (4.10) and (4.12) by the combined model $\hat{\mathbf{X}}^{\text{tot}}$.

(4) Update $\mathbf{g}_{j,t}^{\text{freq}}$ using either (4.16) or (4.19) depending on the chosen cost function. The model $\hat{\mathbf{X}}$ in (4.16) and (4.19) is replaced by $\hat{\mathbf{X}}^{\text{tot}}$.

(5) Evaluate the combined model $\hat{\mathbf{X}}^{\text{tot}}$ (4.25).

(6) Update $\mathbf{B}_{\tau}^{\text{spec}}$ using either (4.11) or (4.13) depending on the chosen cost function. The model $\hat{\mathbf{X}}$ in (4.11) and (4.13) is replaced by $\hat{\mathbf{X}}^{\text{tot}}$.

(7) Update $\mathbf{b}_{j}^{\text{freq}}$ using either (4.18) or (4.20) depending on the chosen cost function. The model $\hat{\mathbf{X}}$ in (4.18) and (4.20) by the combined model $\hat{\mathbf{X}}^{\text{tot}}$.

(8) Repeat Steps (2) –(7) until the values converge.

Figure 4.7 illustrates an example signal and components of the combined model estimated by minimizing the divergence. The complexity of the model was noticed to cause some difficulties for the parameter estimation: the model tended to present drum sounds with time-varying frequencies, so that the optimization stuck easily into a local minimum far from the optimum. To steer the algorithm to the correct direction, we allowed the basis functions for time-varying frequencies to have non-zero values only at locations which correspond to a perfectly harmonic sound. As the Figure 4.7 illustrates, the estimated gains are less accurate than those presented earlier for separate models.

The model can also be extended to allow a component to have time-varying spectrum and frequency simultaneously [154]. This further increases the number of free parameters so that obtaining good separation results can become more difficult.

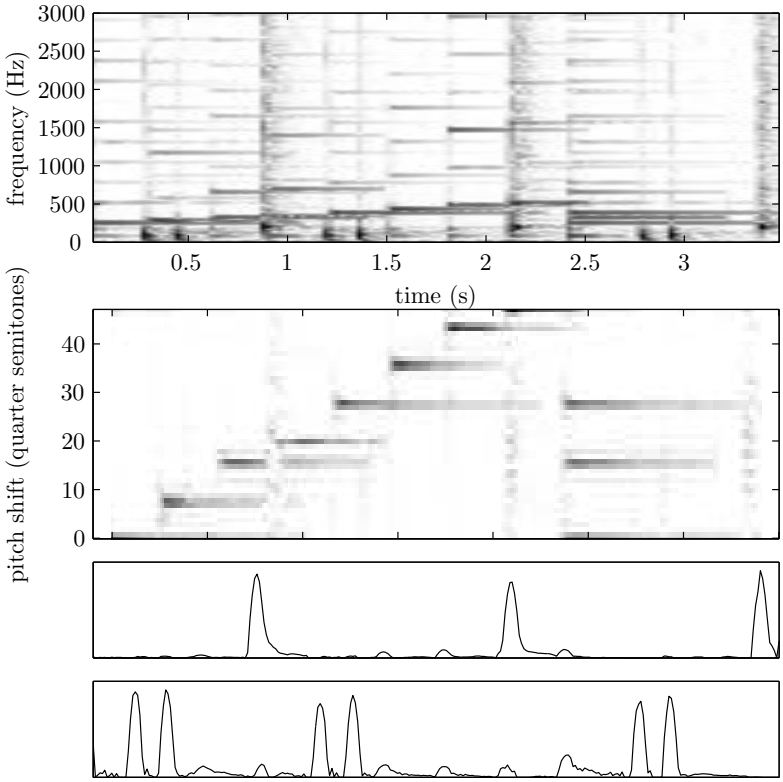

Figure 4.7: Illustration of the model which allows components time-varying spectra and frequencies simultaneously. The mixture signal (upper plot) is a sum of a diatonic scale and C major chord played by guitar (Fig. 2.1) and a drum loop (Fig. 4.2). The mixture signal was separated into a single component with time-varying fundamental frequency (middle plot) and two components with time-varying spectrum (lower plots).

Chapter 5

Overview of Sound Separation Methods Based on Sinusoidal Modeling

An efficient decomposition for the sounds produced by musical instruments is the sinusoids plus noise model, which represents the signal as a sum of deterministic and stochastic parts, or, as a sum of a set of sinusoids plus a noise residual [8, 158]. Sinusoidal components are produced by a vibrating system, and are usually harmonic, i.e. the frequencies are integer multiplies of the fundamental frequency. The residual contains the energy produced by the excitation mechanisms and other components which are not a result of periodic vibration. The deterministic part of the model, which is called sinusoidal model, has been used widely in audio signal processing, for example in speech coding by McAulay and Quatieri [120]. In music signal processing it became known by the work of Smith and Serra [157, 167].

5.1 Signal Model

The sinusoidal model for one frame $x(n),\ n = 0, \ldots, N-1$ of a signal can be written as

$$x(n) = \sum_{h=1}^{H} a_h \cos(2\pi f_h n / f_s + \theta_h) + r(n), \qquad n = 0, \ldots, N-1, \tag{5.1}$$

where n is the time index, N the frame length, a_h, f_h, and θ_h are the amplitude, frequency, and initial phase of the h^{th} sinusoid, respectively, and $r(n)$ is the residual. Most methods assume that the parameters within each frame are fixed, even though time-varying parameters are used in some systems (see for example [65, 125]).

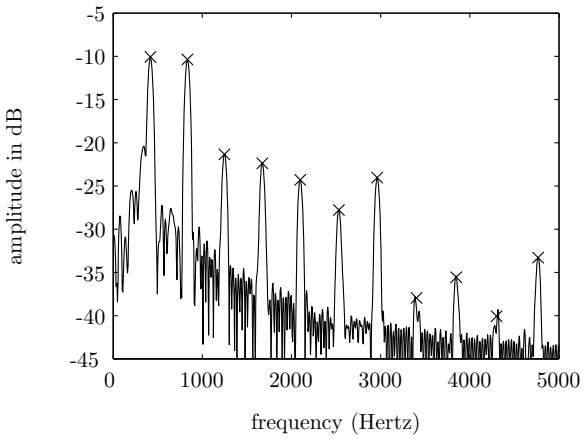

Figure 5.1: The amplitude spectrum of a piano tone G#4. The frequencies (location on the x-axis) and amplitudes (location on the y-axis) of the estimated sinusoids are illustrated with crosses.

Estimation of the parameters is often done in the frequency domain. Each sinusoid corresponds to a peak in the amplitude spectrum, and their frequencies can be estimated by picking the most prominent peaks. Figure 5.1 shows an example of the spectrum of a harmonic sound, from which the sinusoids are estimated. Also matching pursuit algorithms [70] have been used, where a dictionary of time-domain elements is employed to decompose the signal. Basic algorithms for the estimation of sinusoids from music signals have been reviewed by several authors, for example Rodet in [145], Serra in [158], and Virtanen in [187]. A theoretical framework for the estimation is discussed by Kay in [96, pp. 407-445], and many useful practical details on estimating the peaks are given in [6].

The sinusoidal model is a powerful tool in the analysis of music signals, since it can preserve the exact frequencies of the harmonic partials. This enables associating the sinusoids into sound sources and estimating higher-level information such notes played by each source. This approach has been taken, for example, in the fundamental frequency estimation method proposed by Maher and Beauchamp [118]. The amplitudes of the sinusoids have been used, for example, to estimate the parameters for sound synthesis of plucked string instruments [198]. In addition to analysis, the sinusoidal model has also been applied in parametric coding [116, 183].

For a harmonic sound source, the frequencies of the sinusoids are approximately integer multiplies of the fundamental frequency. By taking this into account in the signal model, and by estimating whole harmonic structures instead of individual sinusoids, the robustness of partial estimation can be significantly increased.

For the sum of M sources the signal model can be reformulated as

$$x(n) = \sum_{m=1}^{M} \sum_{h=1}^{H_m} a_{m,h} \cos(2\pi f_{m,h} n + \theta_{m,h}) + r(n), \qquad n = 0, \ldots, N-1 \quad (5.2)$$

where H_m is the number of overtones in source m, and $a_{m,h}$, $f_{m,h}$, and $\theta_{m,h}$ are the amplitude, frequency, and phase of its h^{th} overtone. We assume that the sources are harmonic, so that $f_{m,h} \approx h f_{m,1}$, where $f_{m,1}$ is the fundamental frequency of the source m. For a single monophonic source the signal model was first proposed by Laroche et al. [110], and several authors have extended it for multiple polyphonic sources.

5.2 Separation Approaches

Sound source separation algorithms which apply sinusoidal model can be roughly divided into three categories: 1) methods which first estimate sinusoids and then group them into sound sources, 2) methods which estimate jointly the number of sources, their F0s, and parameters of sinusoids, and 3) methods which first estimate the number of sources, their F0s, and then estimate sinusoids using partial frequencies predicted by the F0s. In addition, we shortly discuss here methods which are based on comb filtering, since their performance is often similar to sinusoidal modeling based methods.

5.2.1 Grouping

The first approach is motivated, for example, by the human auditory system, which has been suggested to use a slightly similar approach in auditory scene analysis. The acoustic cues used by the auditory system listed on Page 5 can be used to design rules for grouping the sinusoids to their sources. Sound source separation algorithms based on this approach have been proposed by Kashino [94, 95], Abe and Ando [5], Sterian [169], and Virtanen [185].

Most pitched musical instruments are harmonic, and often the harmonicity assumption alone is sufficient to perform the grouping. The first source separation algorithms based on the harmonicity limited to two sources. The method proposed by Parsons [134] first estimated individual sinusoids, then estimated the pitches of both sound based on these, and finally assigned each sinusoid to either source by predicting the overtone frequencies using the pitch estimates. Also the system proposed by Maher first estimated the frequencies of the sinusoids and then estimated the pitches based on these [119].[1]

Separation algorithms based on the grouping approach are likely to produce good results only in simple mixing cases. When the number of concurrent sounds is high, a sinusoid may be the result of two or more overlapping sources,

[1] The mismatch between the estimated sinusoids and the overtones predicted using the estimated pitches was used in multiple fundamental frequency estimation by Maher and Beauchamp in [118].

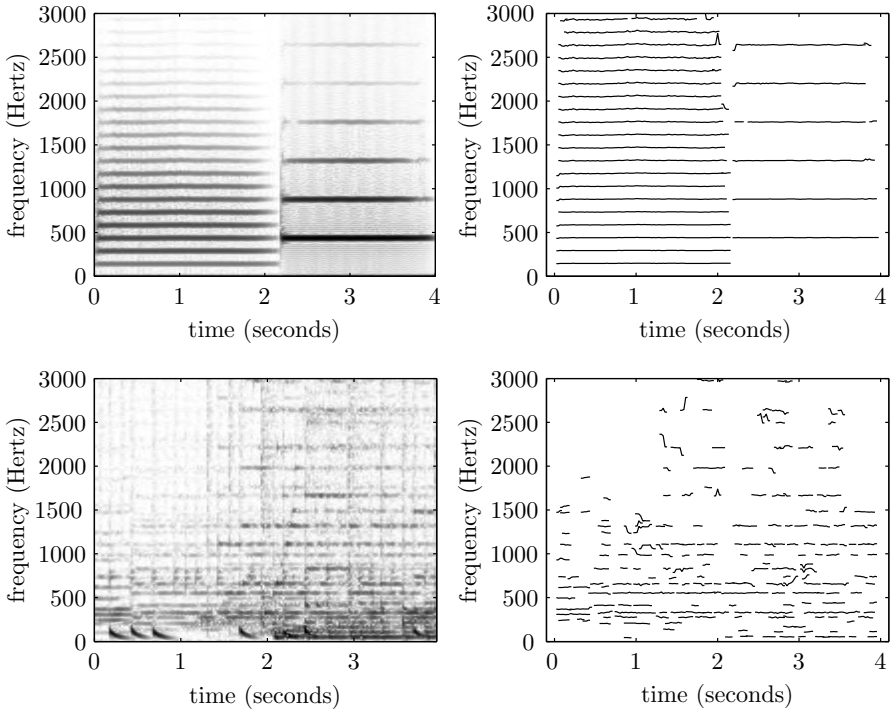

Figure 5.2: The spectrogram of two tones played by French horn successively (upper left panel), a polyphonic excerpt of pop music ('Being Boring' by Pet Shop Boys, lower left panel), and the frequencies of the sinusoids analyzed from both signals (upper and lower right panels). The harmonic structure of the upper signal is clearly visible and the sinusoids can be estimated reliably, whereas the polyphonic signal is more difficult to analyze.

and therefore cannot be resolved only by grouping. Figure 5.2 illustrates the difference between a simple monophonic signal and a polyphonic signal.

5.2.2 Joint Estimation

In practice, estimation of overlapping partials requires that the harmonic structure is utilized already in the estimation of the sinusoids. This can be accomplished by estimating all the parameters in (5.2) jointly, i.e., estimating the parameters of the sinusoids, while restricting them to harmonic relationships.

The estimation criterion can be the modeling error, for example. Since it can be minimized by using an indefinite number of sinusoids, their number has to be limited and parameters restricted so that they are likely to model individual sound sources. For example, harmonic structures can be retained by the constraint $f_{m,h} = hf_{m,1}$. More flexible algorithms, however, do not use strict

constraints, but allow some deviation from the ideal harmonic frequencies. One possibility for formulating flexible restrictions is by a Bayesian framework: prior information is described using probability density functions, which may allow slight deviations from ideal harmonicity, and then Bayesian inference is used to estimate the parameters by maximizing their posterior pdf. Most commonly used prior probability density functions assume a small number of sources, and partial frequencies which are approximately integer multiples of the fundamental frequency.

The Bayesian approach has been used for example by Goto for the detection of the melody and bass lines in polyphonic music [67] and by Godsill and Davy for multiple fundamental frequency estimation [63]. Also the parametric audio coding algorithm proposed by Vincent and Plumbley [183] used Bayesian inference to estimate the signal model parameters.

In the Bayesian framework the posterior pdf is irregular and sharply peaked as a function of fundamental frequencies. Therefore, lot of work is required to explore the parameter space efficiently so as to find the maximum of the posterior pdf [38]. There are several studies which concentrate only on designing a suitable optimization algorithm, see for example [200] and [9]. In practice, a good performance is obtained by initializing the optimization algorithm by an estimate of the fundamental frequencies, which results in fundamental frequency driven estimation.

5.2.3 Fundamental Frequency Driven Estimation

Fundamental frequency estimation has been one of the most studied audio analysis topics, and recently also several multiple fundamental frequency estimation algorithms have been proposed (see [105] for a review). Compared with the source separation, it can usually been performed more robustly, since unlike source signals, pitch is a one-dimensional feature.

The third class of separation algorithms first applies a multiple fundamental frequency estimation algorithm to estimate the number of sounds and their fundamental frequencies, and then estimates the sinusoids by assuming that their frequencies are approximately integer multiplies of the fundamental frequencies. Compared with joint estimation of the parameters, these lose some flexibility, but are usually easier to implement and are typically faster and more robust. Methods based on this approach have been proposed by Quatieri and Danisewicz [142], and Virtanen and Klapuri [194,195], who estimated the parameters by minimizing the energy of the residual. Every and Szymanski separated the sources using a time-varying filter for each source, which were designed using an algorithm initialized by multiple fundamental frequency estimation [52].

5.2.4 Comb Filtering

In addition to sinusoidal modeling, relatively similar results can be obtained by comb filtering, where delayed versions of the input (feedforward filter) or output (feedback filter) signal are summed to the output signal [75, pp. 361-376].

The frequency response of a comb filter has peaks at integer multiplies of the frequency corresponding to the period of the delay. When the delay is tuned according to the fundamental frequency, subtracting delayed versions of the input signal results in canceling the fundamental frequency and its overtones [39].

A normal discrete-time implementation of the delay restricts the fundamental frequencies to quantized values. This problem has been addressed for example by Välimäki et al., who used a fractional-delay allpass filter to achieve arbitrary fundamental frequencies [199]. The method proposed by Wang overcomes the quantization by modulating the signal around the frequency of each sinusoid to the baseband [201]. Furthermore, the algorithm is able to follow the temporal evaluation of the strongest harmonic structure by adaptively updating the modulation frequencies. The fundamental frequency and amplitude envelopes are restricted to be slowly-varying, so that the resulting signals are similar to those obtained using frame-by-frame sinusoidal modeling. In our simulations, it produced good results when a single source was dominating.

5.3 Resolving Overlapping Overtones

Source signals which overlap each other in time and frequency are difficult to resolve with any separation algorithm. In the case of sinusoidal modeling, harmonic fundamental frequency relationships are particularly difficult. When two or more sources are in a harmonic relationship, some of the overtones have approximately the same frequency and these partials are said to collide, or, overlap. The phenomenon is common in musical signals, since in music the fundamental frequencies are often in a harmonic relationship. The exact amplitudes and phases of colliding partials cannot be solved based on the acoustic information, but they can be only approximated.

The rough shape of the amplitude spectrum of natural sounds is usually slowly-varying as a function of time and frequency.[2] It has also been observed that this spectral smoothness principle is an important cue in fusing spectral components into sound sources [22, p. 232]. A possible reason for this is that we do not perceive individual partials, but harmonic structures within frequency bands. For example, it has been observed by Klapuri that for harmonic sounds where the overtones have uniform amplitudes within a frequency band the auditory model [121] produces representations with a large energy at the fundamental frequency [101, pp. 238-241]. Because of the above-mentioned acoustic properties and human sound perception, the amplitudes of overlapping partials can be rather well approximated by interpolating from adjacent frames or partials.

For some musical instruments the overtones are phase-locked [61, pp. 143-144], so that interpolation could also be used to solve the phases of overlapping partials. However, phase locking cannot be assumed in general, since there are also musical instruments where this does not hold [61, pp. 144]. Since the phases are perceptually less important, a careful estimation of the phases is

[2] Klapuri calls this "spectral smoothness", see [106, pp. 54-55].

usually not necessary. Either the phase of the mixture signal can be assigned for all the overlapping harmonics, or their phases can be generated to produce smooth transitions between frames.

The most commonly used method to approximate the amplitude $a_{m,h}$ of the h^{th} partial of source m is by linear interpolation of two adjacent partials $h-1$ and $h+1$, so that

$$\hat{a}_{m,h} = \frac{1}{2}(a_{m,h-1} + a_{m,h+1}). \tag{5.3}$$

This method has been used, for example, by Parsons [134] and Maher [119]. Even though other partials could be used too, the adjacent ones usually produce the best results, since their amplitudes correlate the most with the amplitude of the overlapping partial. Interpolation from adjacent frames has been used in speech separation by Quatieri and Danisewicz [142]. For musical signals their method is not as useful since in music, partials usually collide for a relatively long duration.

By regarding a group of colliding sinusoids as an individual sinusoid, we can measure its amplitude a_{mix}, frequency, and phase. The interpolated amplitudes of each source can be further processed so that the processed sinusoids result in a mixture sinusoid with amplitude a_{mix}. This can be obtained by normalization

$$\tilde{a_i} = a_{\text{mix}} \frac{\hat{a_i}}{\sum_{j=1}^{J} \hat{a}_j}, \tag{5.4}$$

where $\hat{a}_j$, $j = 1, \ldots, J$ are the amplitudes of the interpolated sinusoids. The spectral filtering method proposed by Every and Szymanski [52] implements this principle.

The accuracy of the interpolation depends notably on the accuracy of the fundamental frequency estimation, since that is used to determine which partials are interpolated. Because multiple fundamental frequency estimation is a difficult task, the detection or fundamental frequency estimation of a harmonic sound may fail. As a results of this, the signal $x(n)$ contains harmonic components for which no sinusoids has been assigned in the model (5.2), and some overlap estimations are incorrect. Errors in the fundamental frequency estimation cause large errors in the estimation of partials, for example because interpolating amplitudes from partials which are overlapping. As the number of concurrent sources increases, the number of overlapping partials increases, and there may be situations where most of them overlap. Therefore, there should be a method to utilize also the amplitudes measured for the overlapping component groups. The methods proposed in the next chapter provides one alternative for this.

A high-energy noise residual will also affect the estimation. Non-harmonic musical sources, such as drums, are usually short in duration, so that the energy is high in a few frames only. Non-harmonic noise can be suppressed by using standard noise reduction techniques such as the spectral subtraction [179, pp. 333-354]. It is also possible to suppress noise at a post-processing stage, as proposed by Jensen and Hansen [91]. However, noise reduction is not discussed in this thesis.

Chapter 6

Proposed Separation Method Based on Sinusoidal Modeling

This chapter proposes a separation algorithm based on the sinusoidal model presented in the previous chapter. To achieve computational efficiency and easy implementation, we propose a two-stage method, where first the number of sounds and their rough fundamental frequencies are estimated, after which we estimate their time-varying parameters and perform the actual separation. For resolving overlapping overtones we propose and test several methods, which all are based on the spectral continuity assumption discussed in Section 5.3.

In the first stage we apply the multiple fundamental frequency estimator [105] which has been shown to produce good results in this task. The separation and estimation of more accurate sinusoidal modeling parameters is done by minimizing the least-squares error between the mixture signal and the model, while using restrictions which accomplish the separation. Even though the Bayesian approach would provide a greater flexibility than the proposed deterministic algorithm, there are no extensive statistical measurements for the prior distributions functions of the parameters which are required in the Bayesian approach.

Using the same notation for the variables as in the previous chapter, the model $\hat{x}(n)$ for M sources is written as

$$\hat{x}(n) = \sum_{m=1}^{M} \sum_{h=1}^{H_m} a_{m,h} \cos(2\pi f_{m,h} n + \theta_{m,h}), \qquad n = 0, \ldots, N-1, \tag{6.1}$$

When multiple sources ($M > 1$) are present, each of them is modeled with a separate part of the signal model. Minimization of the reconstruction error between the observed signal $x(n)$ and the model $\hat{x}(n)$ does not in general guarantee that each source signal would be represented separately with its own

part of the model. For example, if two overtones have the same frequency, either of them can represent the signal energy at that frequency. Therefore, we place restrictions for the model parameters so that it becomes unlikely that a source is represented with another source's part. In practice, the amplitudes of overlapping partials are restricted so that the large changes between adjacent overtones of a sound are not allowed, resulting in continuous spectrum of separated sounds.

If the residual in (6.1) is assumed to be normally distributed white noise, least-squares estimation is the maximum likelihood estimator for individual sinusoids. Nonlinear least-squares (NLS) algorithm can be used to estimate their frequencies, amplitudes, and phases [170]. Stoica and Nehorai found out, that in colored noise, frequency estimates of NLS are the same as in the case of white noise [171]. In colored noise, the amplitudes have to be adjusted by bandwise noise subtraction. In the estimation of closely spaced sinusoids the NLS has been used by Tolonen in [173] and by Virtanen in [187], and in the estimation of perfectly harmonic sounds by Karjalainen and Tolonen [92, 174].

Since the NLS method itself does not provide a measure for determining the number of sounds, we estimate it by the multiple fundamental frequency estimator (MFFE) proposed by Klapuri in [105]. The MFFE produces also rough estimates of the overtone frequencies which can be used to initialize the NLS algorithm. The estimated F0s are used to generate rough predictions of the partial frequencies, which are then improved iteratively. We use an extended version of the method proposed by Depalle and Tromp [43] to estimate the sinusoids. Overlapping components are resolved using linear models for the overtone series or by nonlinear smoothing, as described in Section 6.3.

The overall separation procedure consists of the following steps:

(1) Detect onset times.

(2) Estimate multiple fundamental frequencies between adjacent onsets.

(3) Select a linear model based on the fundamental frequency estimates.

(4) Estimate phases, keeping the frequencies fixed (Section 6.2).

(5) Estimate amplitudes, keeping the phases and frequencies fixed (Sect. 6.3).

(6) Update frequencies, keeping the amplitudes and phases fixed (Section 6.4).

(7) Iterate steps 4 - 6.

(8) Combine the estimated fundamental frequencies in successive frames to form entire notes (Section 6.5).

The steps 4 - 6 are our main contribution in this separation algorithm, so they are covered in more detail in the following sections. The steps 1 and 2 are shortly described here.

The onset detection (Step 1) estimates the temporal locations of sound onsets. We use the algorithm of Klapuri [99], which measures the amplitude envelope of the signal within 21 non-overlapping frequency bands, estimates onsets

within each band by locating large relative increases in the amplitude envelope, and then combines the onsets from all the bands.

We assume that sounds set on and off exactly at the estimated onset locations, so that between two onsets the number of sounds and their fundamental frequencies do not change (more accurate onset and offset locations can be estimated from the sinusoids). In Step 2 we estimate the number of sounds and their fundamental frequencies by the MFFE estimator proposed by Klapuri in [105]. In addition to the fundamental frequency estimates, the algorithm produces rough estimates of the overtones frequencies. Because the algorithm uses a relatively long window size (90-200 ms), it can estimate the number of sounds and their fundamental frequencies quite robustly.

Since onsets of other instruments can occur during a musical note, a note can be divided into several adjacent MFFE frames. Once the time-varying sinusoidal modeling parameters have been estimated (Steps 4 - 6), parameters estimated in successive frames are combined to form notes (Step 8), which is shortly described in Section 6.5.

6.1 Formulation in the Frequency Domain

The proposed method can be formulated both in time and frequency domains. Here we present the frequency-domain formulation, since it allows a computationally efficient estimation of the phases by processing the real and imaginary parts of the spectrum separately. Furthermore, each sinusoid is localized in the frequency domain, and therefore a narrow frequency band is sufficient for the estimation of its parameters. This leads to an approximation which reduces the computational complexity significantly.

This section presents how the parameters of (6.1) are estimated within each short sinusoidal modeling frame, given the rough frequency estimates $\hat{f}_{m,h}$. Frame sizes between 15 and 100 ms were tested in our simulations, but a frame size 40 ms was used in the final system, since it provides a good tradeoff between time and frequency resolutions. A 50% overlap between adjacent frames was used. We used the Hamming [17, pp. 95-98] window. The sidelobeless window functions proposed by Depalle and Hélie [42] could ideally provide better results if the initial frequency estimates are far from the correct ones, since their responses are monotonically decreasing. However, here the initial frequency estimates are rather close to the correct values, and therefore the mainlobe's width is more important, and the results were better with the Hamming window (see Sec. 6.6).

To enable phase estimation, the model is rewritten as

$$x(n) = \sum_{m=1}^{M} \sum_{h=1}^{H_m} \alpha_{m,h} \cos(2\pi f_{m,h} n) + \beta_{m,h} \sin(2\pi f_{m,h} n), \tag{6.2}$$

where $\alpha_{m,h} = a_{m,h} \cos(\theta_{m,h})$ and $\beta_{m,h} = -a_{m,h} \sin(\theta_{m,h})$.

The frequency transform of the model is developed so that the spectrum of the cosine terms is real and the spectrum of the sine terms is imaginary. In practice this requires changing the phase basis of the Fourier transform. The main benefits of the operation are lower memory consumption (the terms are either real or imaginary), and lower computational complexity (the parameters can be solved separately for the real and imaginary parts).

The Fourier transform of a real-valued signal which is symmetric with respect to the origin is also real [21, pp. 14-15]. As suggested by Harris [74], the frequency transforms are here calculated so that the time index $n = N/2$ (T being the frame length in samples) is regarded as the origin. As a result of this, the window function and the cosine terms become even. This technique is often referred as "zero-phase windowing", and in practise it can be implemented by using the normal FFT and subtracting a linear phase term $\pi Nk/K$ from the resulting phase spectrum. Thus, the zero-phase DFT is otherwise equal to the normal DFT but the phase of the basis is changed.

After this processing, the DFT of each cosine term multiplied by the window function is real, and the DFT of a sine term (odd function) multiplied by $w(n)$ is imaginary. Particularly, according to the modulation theorem [21, p. 108], the Fourier transform of $w(n)\cos(f_0 n)$ equals $\frac{1}{2}[W(f-f_0)+W(f+f_0)]$, and the Fourier transform of $w(n)\sin(f_0 n)$ equals $\frac{i}{2}[W(f-f_0)-W(f+f_0)]$, where i is the imaginary unit and $W(f)$ is the Fourier transform of $w(n)$.

By applying the above result on (6.2), we obtain the frequency-domain model for $X(f)$, the Fourier transform of windowed frame $x(n)$. The real and imaginary parts, $\Re\{X(f)\}$ and $\Im\{X(f)\}$, can be written separately as

$$\begin{aligned}\Re\{X(f)\} &= \sum_{m=1}^{M}\sum_{h=1}^{H_m}\alpha_{m,h}H_{m,h}^{\Re}(f)\\ \Im\{X(f)\} &= \sum_{m=1}^{M}\sum_{h=1}^{H_m}\beta_{m,h}H_{m,h}^{\Im}(f),\end{aligned} \tag{6.3}$$

where the scalars $\alpha_{m,h}$ and $\beta_{m,h}$ are real,

$$H_{m,h}^{\Re} = \frac{1}{2}[W(f-f_{m,h})+W(f+f_{m,h})] \tag{6.4}$$

is real, and

$$H_{m,h}^{\Im} = \frac{i}{2}[W(f-f_{m,h})-W(f+f_{m,h})] \tag{6.5}$$

is imaginary, for all $m = 1,\ldots,M$, and $h = 1,\ldots,H_m$.

6.2 Phase Estimation

On the first iteration the phases are estimated by using the frequency estimates $\hat{f}_{m,h}$ given by the MFFE, but on later iterations the refined (see Sec. 6.4) overtone frequencies are used.

As discussed earlier (see Section 5.3), there is no general method for the estimation of the phases of overlapping components. Since the phases are not perceptually as important as the amplitudes and frequencies, we simply set the same phase for all the overlapping components.

Firstly, overlapping components are detected by finding groups of sinusoids, the frequencies of which are within a predefined threshold. In our system, the threshold $0.5 f_s/N$ was found to be good (the natural resolution of the spectrum is f_s/N, but the least-squares method utilizes also adjacent frequency lines, which increases its accuracy). In the phase estimation, each group is regarded as a single sinusoid, the frequency of which is the mean of the overlapping frequencies. Let us denote the total number of sinusoids (non-overlapping ones plus each group) by J.

Let us write the basis functions in (6.4) and (6.5) by two K-by-J matrices $\mathbf{H}_\Re$ and $\mathbf{H}_\Im$, which contain the real part and the imaginary part of the sinusoids' spectra, respectively. Each column of the matrices is the DFT of an individual sinusoid evaluated at discrete frequencies $k = 1, \ldots, K$. The window length 40 ms with sampling frequency 44100 Hertz corresponds to $N = 1764$. We used zero-padding and FFT length of 8192 but the sinusoids were estimated and FFT bins used only up to 5000 Hz, which is the first $K = 929$ FFT bins. We implemented the evaluation of the DFTs by calculating $W(k)$ using a high frequency resolution, and then choosing the indices corresponding to each sinusoid.

The model for the spectrum vector $\mathbf{x} = \big[X(1), \ldots, X(K)\big]^\mathsf{T}$ can be written separately for its real part $\mathbf{x}_\Re$ and imaginary part $\mathbf{x}_\Im$ as

$$\begin{aligned} \mathbf{x}_\Re &= \mathbf{H}_\Re \boldsymbol{\alpha} \\ \mathbf{x}_\Im &= \mathbf{H}_\Im \boldsymbol{\beta}, \end{aligned} \tag{6.6}$$

where $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$ are vectors, the j^{th} element of which is related to the amplitude and phase of the j^{th} sinusoid as $\alpha_j = a_j \cos(\theta_j)$ and $\beta_j = -a_j \sin(\theta_j)$.

Because the real and imaginary parts are orthogonal, least-squares solution for their parameters can be solved separately. The least squares solution [97] for (6.6) is

$$\begin{aligned} \hat{\boldsymbol{\alpha}} &= \left(\mathbf{H}_\Re^\mathsf{T} \mathbf{H}_\Re\right)^{-1} \mathbf{H}^\mathsf{T} \mathbf{x}_\Re \\ \hat{\boldsymbol{\beta}} &= \left(\mathbf{H}_\Im^\mathsf{T} \mathbf{H}_\Im\right)^{-1} \mathbf{H}^\mathsf{T} \mathbf{x}_\Im. \end{aligned} \tag{6.7}$$

The rows of $\mathbf{H}_\Re$ and $\mathbf{H}_\Im$ are linearly independent if two or more sinusoids do not have equal frequencies. Since such cases were detected and grouped previously, the matrices have full rank, and the inverses in (6.7) exist.

From the above solution, the phase θ_j of the j^{th} sinusoid is obtained as the phase of the complex variable $(\alpha_j + i\beta_j)$. If components are not overlapping, also the least squares solution for the amplitudes can be obtained as $\hat{a}_j = \sqrt{\alpha_j^2 + \beta_j^2}$

The above solution for K-by-J $\mathbf{H}_\Re$ and $\mathbf{H}_\Im$ requires $(K + J/3)J^2$ floating point operations (flops) for both $\hat{\boldsymbol{\alpha}}$ and $\hat{\boldsymbol{\beta}}$ [64, pp. 236-247], making $2(K+J/3)J^2$ flops in total.

In the estimation of sinusoidal parameters, previous methods (for example [97, pp. 255-257], [42]) solve the real and imaginary parts (or the in-phase and quadrature-phase parts [170]) simultaneously, requiring $4(K + 2J/3)J^2$ flops. Thus, solving the real and imaginary parts separately reduces the computational complexity by factor $2 \ldots 4$, depending on ratio of K and J. Furthermore, matrices in the proposed method are either real or imaginary, requiring a smaller number of additions and multiplications compared to methods which operate on complex variables [140, pp. 176-178].

The computational complexity of the least-squares solutions (6.7) can be further reduced considerably by an approximation where the basis functions are grouped into subsets according to their frequencies, and the parameters are solved separately for each subset. In practice the approximation produces accurate results, since each basis function is localized in the frequency, so that basis functions with significantly different frequencies are approximately orthogonal.

6.3 Amplitude Estimation

This section presents two alternative methods for the amplitude estimation of overlapping components, both of which are based on the spectral continuity assumption. The first one, which uses linear models for the overtone series was originally proposed in [195], and is here presented with some extensions. The second one which uses nonlinear smoothing was originally proposed by Klapuri in [100] and incorporated to sinusoidal modeling by Virtanen and Klapuri in [194].

The linear model for the overtone series includes the spectral smoothness principle in the core signal model, and it guarantees that the sum of the amplitudes of overlapping sinusoids equals the amplitude measured for the group of sinusoids. Ideally, this makes it more robust than methods which post-process the estimated amplitudes.

The overtone amplitudes of a sound are modeled as a weighted sum of fixed basis functions. Let us denote the overtone amplitudes of a source m by a vector $\mathbf{a}_m = \left[a_{m,1}, \ldots, a_{m,H_m}\right]^\mathsf{T}$. For source m, the linear basis is written using a H-by-L matrix $\mathbf{G}_m$, where $H \geq L$. Each column of the matrix is one basis function. The l^{th} entry of vector $\mathbf{u}_m$ is the weight of the l^{th} basis function, so that the model for the amplitudes is written as

$$\mathbf{a}_m = \mathbf{G}_m \mathbf{u}_m \tag{6.8}$$

The model can also be viewed so that the amplitudes are estimated in a subspace which does not allow large changes between the amplitudes of adjacent overtones.

First we present how the amplitudes can be solved given a certain $\mathbf{G}_m$, and then discuss different model types.

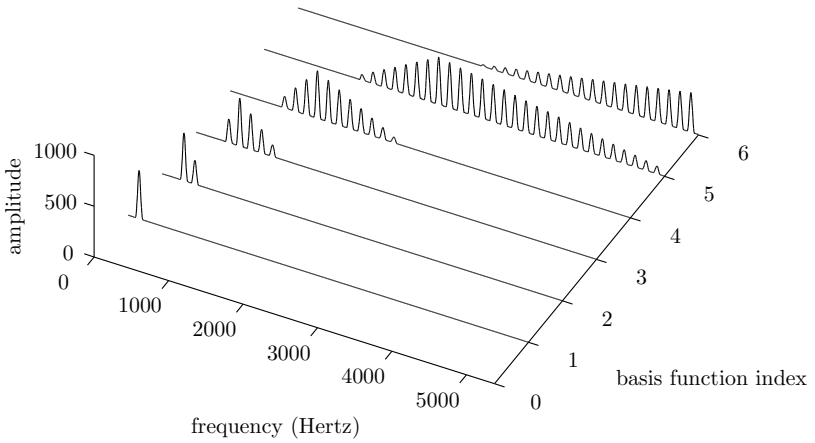

Figure 6.1: A linear frequency-band model for the spectrum of a harmonic sound, where the basis function (columns of $\mathbf{Y}_m$) are harmonic combs .

6.3.1 Least-Squares Solution of Overlapping Components

Based on the modulation theorem and Euler formula, the Fourier transform of the windowed cosine $w(n)\cos(2\pi f_{m,h}n + \theta_{m,h})$ in (6.1) can be written by

$$H_{m,h}(f) = \frac{1}{2}[e^{i\theta_{m,h}}W(f - f_{m,h}) + e^{-i\theta_{m,h}}W(f + f_{m,h})]. \tag{6.9}$$

Using the current frequency and phase estimates, these are written into a K-by-H_m matrix $\mathbf{H}_m$ for each source m. The h^{th} column of $\mathbf{H}_m$ is the spectrum of the h^{th} sinusoid of source m evaluated at discrete frequencies $k = 1, \ldots, K$. The model for the spectrum of source m is then $\mathbf{H}_m\mathbf{a}_m$, and when we apply the linear model from (6.8), the model can be written as $\mathbf{H}_m\mathbf{G}_m\mathbf{u}_m$. Let us denote $\mathbf{Y}_m = \mathbf{H}_m\mathbf{G}_m$, so that the model equals $\mathbf{Y}_m\mathbf{u}_m$. Now the spectrum of source m is modeled as a weighted sum of the columns of $\mathbf{Y}_m$. With a suitable linear model $\mathbf{Y}_m$ the columns can be, for example, harmonic combs, as illustrated in Figure 6.1. A linear model consisting of harmonic combs within frequency bands implements an estimation principle which corresponds roughly to the human perception of harmonic sounds, as discussed in Section 5.3.

For M sources the matrices are combined as $\mathbf{Y} = \left[\mathbf{Y}_1, \ldots, \mathbf{Y}_M\right]$. Let us denote the gains of all the sources using vector $\mathbf{u}^\mathsf{T} = \left[\mathbf{u}_1^\mathsf{T}, \ldots, \mathbf{u}_M^\mathsf{T}\right]$, so that the model for the spectrum $\mathbf{x}$ equals $\mathbf{x} = \mathbf{Yu}$. Depending on the linear model, the gains are restricted to either real or non-negative values. The restrictions can be implemented by rewriting the model as $\mathbf{x}_{\Re\Im} = \mathbf{Y}_{\Re\Im}\mathbf{u}$, where $\mathbf{x}_{\Re\Im} = \begin{bmatrix}\mathbf{x}_\Re \\ \mathbf{x}_\Im/i\end{bmatrix}$,

and $\mathbf{Y}_{\Re\Im} = \begin{bmatrix} \mathbf{Y}_\Re \\ \mathbf{Y}_\Im / i \end{bmatrix}$, $\mathbf{Y}_\Re$ and $\mathbf{Y}_\Im$ being the real and imaginary parts of $\mathbf{Y}$, respectively. Now the least-squares solution for real-valued gains is obtained as

$$\hat{\mathbf{u}} = \left(\mathbf{Y}_{\Re\Im}^{\mathsf{T}} \mathbf{Y}_{\Re\Im}\right)^{-1} \mathbf{Y}_{\Re\Im}^{\mathsf{T}} \mathbf{x}_{\Re\Im}. \tag{6.10}$$

When the gains are restricted to non-negative values, the least-squares solution to $\mathbf{x}_{\Re\Im} = \mathbf{Y}_{\Re\Im}\mathbf{u}$ is obtained using the non-negative least-squares algorithm [111, p. 161].

With all the models, the amplitudes are solved from the resulting $\hat{\mathbf{u}}_m$ by $\hat{\mathbf{a}}_m = \mathbf{G}_m \hat{\mathbf{u}}_m$. To reduce the computational complexity, some models (e.g. the frequency-band model) allow an approximation where the gains are solved for a subset of basis functions at a time, as done in the phase estimation at the end of Sect. 6.2.

Fixed polynomial model A simple example of a linear model is the Vandermonde matrix $[\mathbf{G}_m]_{h,l} = f_{m,h}^{l-1}$, which models the amplitudes of the overtone series as a $(L-1)^{\text{th}}$-order polynomial. The results obtained with this model were not particularly good. The main reason for this is that polynomial basis functions have most of their energy at high frequencies, whereas audio spectra typically have a low-pass structure. Another drawback of the polynomial model is that each basis function covers the whole frequency range, while it would be advantageous to use only the adjacent harmonics in the interpolation since they correlate the most with the overlapping ones.

The low-pass characteristics can be better modeled using $[\mathbf{G}_m]_{h,l} = f_{m,h}^{-l+1}$, which models the amplitudes as a polynomial with negative powers. With this model we obtained good results by using $L = \lceil \log_2(H_m) \rceil$. In the case of the polynomial model it is natural to allow the gains to have also negative values.

Fixed frequency-band model In comparison to the polynomial model, better results were obtained using a frequency-band model, where each basis function is zero outside a frequency band. We use triangular frequency bands, which are determined by a set of center frequencies f_l, $l = 1, \ldots, L$. For a particular overtone with frequency $f_{m,h}$, the the h^{th} row of the linear model matrix is given by

$$[\mathbf{G}_m]_{h,l} = \begin{cases} (f_{m,h} - f_{l-1})/(f_l - f_{l-1}), & f_{m,h} \geq f_{l-1} \wedge f_{m,h} < f_l \\ (f_l - f_{m,h})/(f_l - f_{l+1}), & f_{m,h} > f_l \wedge f_{m,h} < f_{l+1} \\ 0, & \text{otherwise.} \end{cases} \tag{6.11}$$

Figure 6.2 illustrates the basis functions obtained using center frequencies $f_l = 100 \times 2^{l-1}$ Hz, $l = 1, \ldots, 5$, which result in octave bands.

Triangular basis functions result in an amplitude spectrum, which is piecewise linearly interpolated, as illustrated in the left panel in Figure 6.3. As the rough spectral shape of natural sounds often decreases as a function of frequency, a small improvement can be obtained by applying $1/f$ scaling on the elements

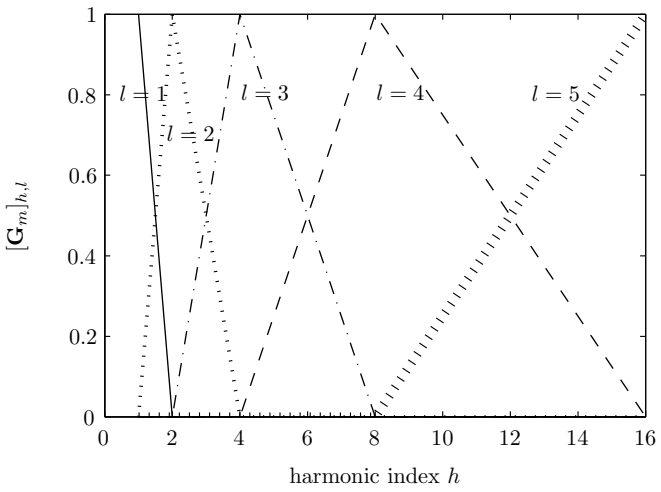

Figure 6.2: The basis functions of a perfectly harmonic sound of 100 Hz fundamental frequency obtained using center frequencies $f_l = 100 \times 2^{l-1}$ Hz, $l = 1, \ldots, 5$. Each basis function is plotted with a different line style.

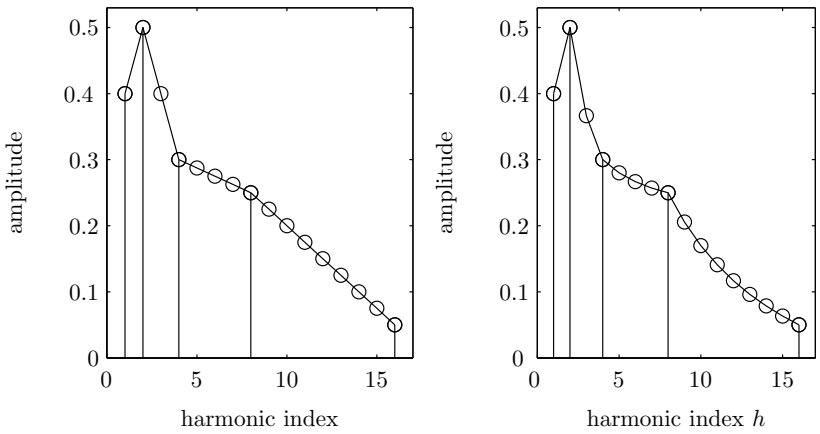

Figure 6.3: Triangular frequency-band model results in a piecewise linear amplitude spectrum (left panel). The basis functions can also be weighted by $1/f$ rough spectral shape (right panel), which corresponds better to natural sounds.

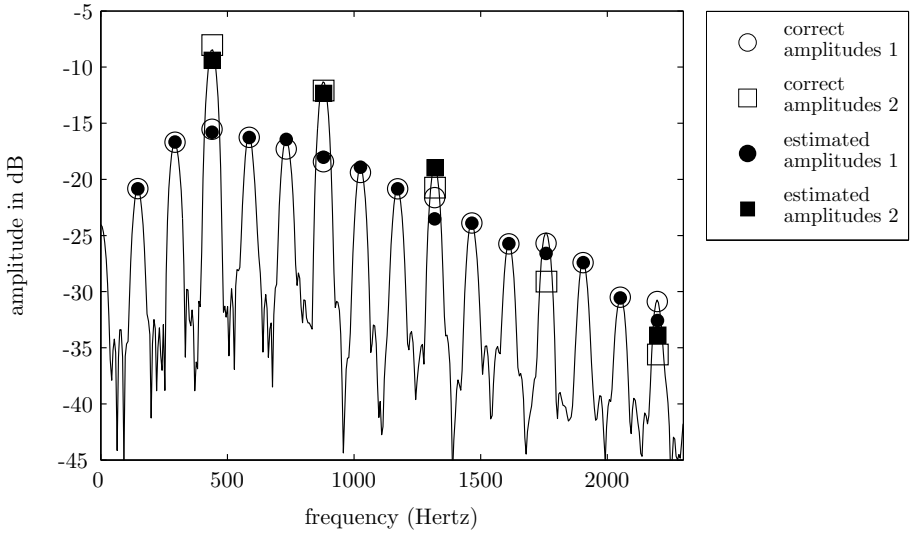

Figure 6.4: The amplitude spectrum of simultaneous tones D3 and A4 played by French horns. The amplitudes of both the partials were estimated using the fixed linear model depicted in Figure 6.3. The estimated amplitudes are plotted with filled circles and boxes, and the correct amplitudes estimated from original unmixed signals with empty circles and boxes, respectively. The third harmonic of the higher sound at about 1750 Hz is missing because the separation algorithm assigned it a zero amplitude.

of $\mathbf{G}_m$, resulting in an interpolated amplitude spectrum depicted in the right panel of Figure 6.3. In the case of frequency-band model, negative gains of the basis functions do not have a meaningful physical interpretation, and therefore they can be restricted to non-negative values.

Figure 6.4 shows a separation example of two sources. The tones are in a harmonic relation, so that every third overtone of the lower sound overlaps with the higher tone. By using the frequency-band model, the overlapping overtones can be quite well approximated.

We tested several different ways to select the center frequencies. It was noticed that it is advantageous that each of them equals the frequency of an overtone. For completely harmonic sounds, good results were obtained by setting the l^{th} center frequency equal to the frequency of the $\lceil 1.5^{l-1} \rceil^{\text{th}}$ harmonic.

Fixed frequency-warped cosine basis model Representing the shape of spectrum using Mel-frequency cepstral coefficients (MFCCs, [37]) is widely used in the classification of audio signals. MFCCs are computed by taking the cosine transform of the log-amplitude spectrum calculated at a Mel-frequency scale. A linear model can approximate the amount of contribution of each MFCC to the amplitude to each sinusoid. The basis functions of the cosine transform

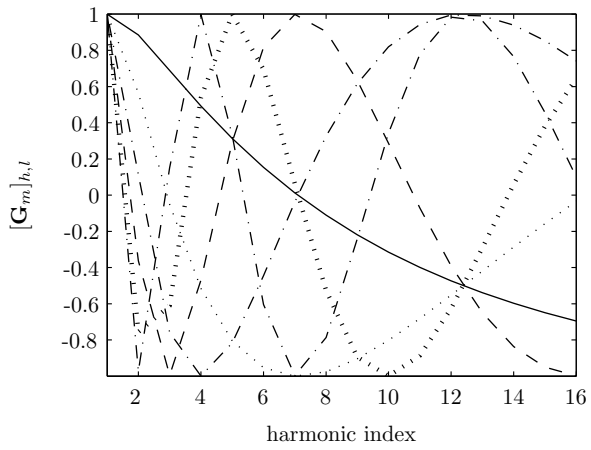

Figure 6.5: Each basis functions of the frequency-warped cosine basis model approximate the amount of contribution of each Mel-frequency cepstral coefficient. The basis functions are cosines, the frequencies of which decrease as a function of the partial index.

are cosines, and the mel-frequency scale can be implemented by warping their frequencies. In the computation of MFCCs the non-linear logarithm operator is applied on the mixture signal, and therefore its contribution to a single basis function cannot be implemented in this framework, and is therefore left out.

The basis functions of the resulting frequency-warped cosine basis model are given as

$$[\mathbf{G}_m]_{h,l} = \cos(l f_{m,h}^{\text{warped}}), \tag{6.12}$$

where $f_{m,h}^{\text{warped}}$ is the warped frequency of the $(m,h)^{\text{th}}$ cosine. We used the warping $f_{m,h}^{\text{warped}} = \log(1 + 10 f_{m,h}/f_{m,1})$. More accurate warpings according to the frequency scales of human hearing are discussed by Härmä et al. in [73].

The resulting basis functions which are illustrated in Figure 6.5 are cosines, the frequencies of which decrease as a function of the partial index. Also with this model we set the number of basis functions to $\lceil \log_2(H_m) \rceil$, and their gains were allowed to have negative values.

Adaptive frequency-band model A fixed model restricts the amplitudes to a subspace of parameters, which may not be able to model a given spectrum exactly. If an overtone is not overlapping, it is not necessary to use the adjacent partials in estimating its amplitude. Non-overlapping components can be more accurately estimated by an adaptive model which interpolates only those components which are overlapping. Moreover, when a group of V overtones is overlapping, and the first $V-1$ of them are approximated by interpolation, the V^{th} overtone can be estimated by using the $V-1$ approximated overtones and the amplitude of the mixture.

We tested several adaptive models, and on the average, the best results were obtained using the following principles to design the model: 1) When a group of V overtones is overlapping, the $V-1$ overtones which are likely to have the lowest amplitudes are interpolated. The amplitudes are predicted based on their harmonic indices: since the energy of the spectrum of natural sounds is distributed approximately according to $1/f$, the overtone with the lowest harmonic index is assumed to have the largest amplitude. From now on, the overtone with the highest predicted amplitude is regarded as a non-overlapping overtone. 2) Each overtone which is not overlapping, is assigned an own basis function, which includes only that particular overtone. 3) Each overtone which is overlapping, is interpolated from the next higher overtone of that sound by including them into the same basis function. If the higher overtone is not overlapping, then the basis includes these two overtones, but if also the higher overtone overlaps, higher overtones are included until one of them is not overlapping. Because of the $1/f$ rough spectral distribution, interpolation from higher overtones is likely to underestimate the amplitude, whereas interpolation from lower overtones is likely to overestimate it. In our simulations, interpolation from higher overtones produced better SDR than interpolation from lower or both adjacent overtones.

A procedure for calculating the basis functions based on the above principles is given as follows: first, we assign an individual basis function for all the sinusoids ($[\mathbf{G}_m]$ is an identity matrix). Then we find a basis function, which fulfills the following two conditions: 1) The highest partial included in the basis function is overlapping and 2) the highest partial does not have the lowest partial index among all the partials within the overlapping group. Once such basis function is found, we sum it to the basis function which contains the immediately higher harmonic of the same sound, and delete the original basis function. Then we find the next basis function which fulfills the above conditions, and repeat until none of the basis functions fulfills them.

Figure 6.6 illustrates the basis functions of the adaptive frequency-band model obtained for the signal in Fig. 6.4. Only those overtones of the lower sound which are overlapping with the higher sound are interpolated. Figure 6.7 illustrates the amplitudes for the two-note signal in Fig. 6.4 estimated using the adaptive frequency-band model. Compared with the fixed frequency-band model in Fig. 6.7, the estimates of non-overlapping components are more accurate.

It is possible to make different variations of the linear models: for example to initialize the adaptive frequency-band model with something else than an identity matrix, or use some other principles in the adaptation. However, in our simulations the above described approach produced the best results among the tested methods. The gains of the frequency-band basis functions were restricted to non-negative values, since there is no meaningful interpretation for negative gains when the basis functions are positive.

Nonlinear smoothing The effect of the overlapping partials, or, harmonic interference, can be decreased by post-processing the estimated amplitudes us-

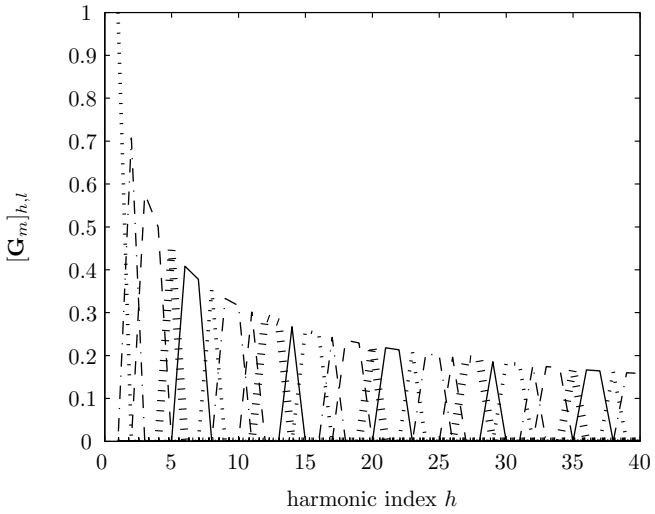

Figure 6.6: The adaptive frequency-band basis functions of the source of a lower fundamental frequency in Fig 6.4. Each overtone of the source having a higher fundamental frequency have their own basis function, since they are estimated to have a higher amplitude.

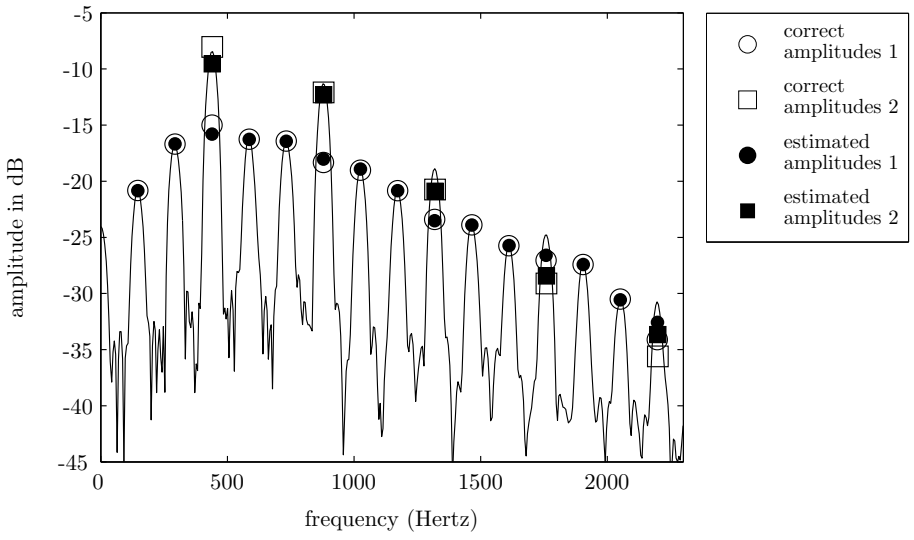

Figure 6.7: Amplitudes of the two-note signal in Fig. 6.4 estimated by the adaptive frequency-band model.

ing perceptually motivated spectral smoothing method proposed by Klapuri in [105]. It calculates a smoothed version of the amplitude spectrum for each source, and if the resulting amplitude of an overtone is below the original amplitude, the amplitude is replaced by the new one. Unlike the other interpolation methods, it does not require knowing the overlapping overtones. It processes all the overtones to obtain better robustness by reducing the accuracy of all the amplitude estimates. A separation algorithm based on the smoothing mechanism was originally proposed by Virtanen and Klapuri in [194].

6.4 Frequency Estimation

On each iteration of the the iterative frequency estimation procedure, we use the current amplitude and phase estimates to linearize the frequency dependence in the vicinity of the current estimates, and then minimize the reconstruction error of the linearized model.

The ratio of each overtone frequency $f_{m,h}$ to the fundamental frequency $f_{m,1}$ is kept fixed, so that the harmonic structure is retained. Thus, we do not update the frequencies of the partials individually, but the whole harmonic structure. In general, the overtone frequency ratios at different fundamental frequencies are not the same [60], but for small frequency deviations within a MFFE frame the approximation is accurate enough.

Presentation here is an extension of [43] and [194]. For each partial, let us write the error between the correct frequency $f_{m,h}$ and the current estimate $\hat{f}_{m,h}$ as $\Delta_{m,h} = \hat{f}_{m,h} - f_{m,h}$. Since the frequency ratios are fixed, the error of each overtone is linearly dependent on the error of the fundamental frequency: $\Delta_{m,h} = \Delta_{m,1} f_{m,h} / f_{m,1}$.

The model for the spectrum is

$$X(f) = \sum_{m=1}^{M} \sum_{h=1}^{H} a_{m,h} H_{m,h}(f) \tag{6.13}$$

where

$$H_{m,h}(f) = \frac{1}{2}[e^{i\theta_{m,h}} W(f - f_{m,h}) + e^{-i\theta_{m,h}} W(f + f_{m,h})]. \tag{6.14}$$

The Taylor series of W at each $f \pm \hat{f}_{m,h}$ is

$$\begin{aligned} W(f \pm \hat{f}_{m,h}) &= W(f \pm f_{m,h} \pm \Delta_{m,h}) \\ &= W(f \pm f_{m,h}) \pm W'_{m,h}(f \pm f_{m,h})\Delta_{m,h} + o(\Delta^2_{m,h}), \end{aligned} \tag{6.15}$$

where W' denotes the derivative of W, and $o(\Delta^2_{m,h})$ includes the terms of second and higher order, which are small in the vicinity of the correct frequency. By discarding $o(\Delta^2_{m,h})$, we obtain the linear approximation

$$W(f \pm f_{m,h}) \approx W(f \pm \hat{f}_{m,h}) \mp W'_{m,h}(f \pm \hat{f}_{m,h})\Delta_{m,h}, \tag{6.16}$$

By substituting (6.16) to (6.14), we get

$$H_{m,h}(f) \approx \hat{H}_{m,h}(f) + \hat{H}'_{m,h}(f)\Delta_{m,h} \tag{6.17}$$

where

$$\hat{H}_{m,h}(f) = \frac{1}{2}[e^{i\theta_{m,h}} W(f - \hat{f}_{m,h}) + e^{-i\theta_{m,h}} W(f + \hat{f}_{m,h})] \tag{6.18}$$

and

$$\hat{H}'_{m,h}(f) = \frac{1}{2}[e^{i\theta_{m,h}} W'(f - \hat{f}_{m,h}) - e^{-i\theta_{m,h}} W'(f + \hat{f}_{m,h})]. \tag{6.19}$$

Let us denote the current spectrum estimate by

$$\hat{X}(f) = \sum_{m=1}^{M} \sum_{h=1}^{H} a_{m,h} \hat{H}_{m,h}(f). \tag{6.20}$$

By substituting (6.17) and (6.20) to (6.13), the approximation for the spectrum becomes

$$X(f) \approx \hat{X}(f) + \sum_{m=1}^{M} \sum_{h=1}^{H} a_{m,h} \hat{H}'_{m,h}(f)\Delta_{m,h} \tag{6.21}$$

Now we can write the error of the spectrum estimate as a function of the fundamental frequency error as

$$X(f) - \hat{X}(f) \approx a_{m,h} \hat{H}'_{m,h}(f) \frac{\hat{f}_{m,h}}{\hat{f}_{m,1}} \Delta_{m,1} \tag{6.22}$$

The modeling error for the discrete spectra can be thus written as

$$\mathbf{x} - \hat{\mathbf{x}} \approx \mathbf{\Omega}\mathbf{\Delta}, \tag{6.23}$$

where $\mathbf{x}$ and $\hat{\mathbf{x}}$ are the observed and modeled spectrum, respectively, and the m^{th} column of the matrix $\mathbf{\Omega}$ is $\sum_{h=1}^{H_m} a_{m,h} \frac{\hat{f}_{m,h}}{\hat{f}_{m,1}} \hat{H}'_{m,h}(f)$ evaluated at discrete frequencies $k = 1, \ldots, K$, and the m^{th} element of $\mathbf{\Delta}$ contains the corresponding error $\Delta_{m,1}$ of the fundamental frequency, which is unknown.

The derivatives $W'(f)$ in the definition (6.19) of the terms $\hat{H}'_{m,h}(f)$ are calculated as follows: the Fourier transform of a signal $nw(n)$ equals $iW'(f)$ [109, p. 277], so get W' as the DFT of signal $-inw(n)$.

The least-squares solution for real-valued fundamental frequency error vector $\mathbf{\Delta}$ is obtained as

$$\mathbf{\Delta} = \left(\mathbf{\Omega}_{\Re\Im}^{\mathsf{T}} \mathbf{\Omega}_{\Re\Im}\right)^{-1} \mathbf{\Omega}_{\Re\Im}^{\mathsf{T}} (\mathbf{x}_{\Re\Im} - \hat{\mathbf{x}}_{\Re\Im}), \tag{6.24}$$

where $\mathbf{\Omega}_{\Re\Im} = \begin{bmatrix} \mathbf{\Omega}_{\Re} \\ \mathbf{\Omega}_{\Im}/i \end{bmatrix}$, $\mathbf{\Omega}_{\Re}$ and $\mathbf{\Omega}_{\Im}$ being the real and imaginary parts of $\mathbf{\Omega}$, respectively, and $\hat{\mathbf{x}}_{\Re\Im} = \begin{bmatrix} \hat{\mathbf{x}}_{\Re} \\ \hat{\mathbf{x}}_{\Im}/i \end{bmatrix}$, $\hat{\mathbf{x}}_{\Re}$ and $\hat{\mathbf{x}}_{\Im}$ being the real and imaginary parts

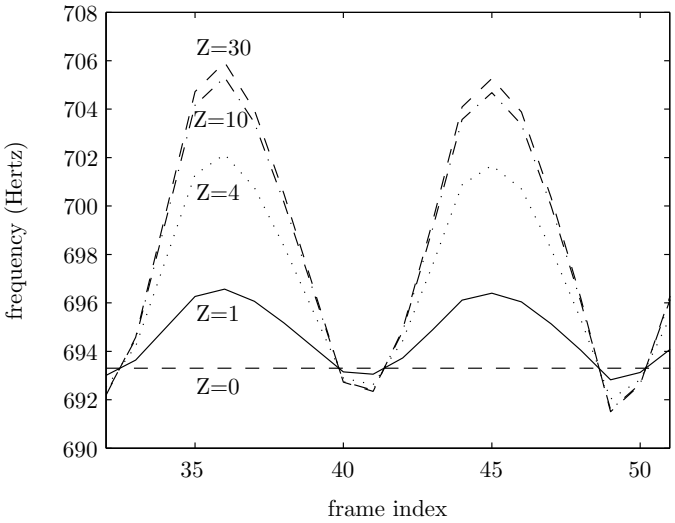

Figure 6.8: The estimated fundamental frequency of an excerpt of a flute signal with different number of iterations Z. The dashed line $Z = 0$ indicates the estimate given by the MFFE. The SDRs for $Z = 0$, 1, 4, 10, and 30 for this signal were 20.1, 22.4, 26.7, 28.3, and 28.4 dB, respectively.

of the spectrum model, respectively. New frequency estimates for each partial are given as $\tilde{f}_{m,h} = \hat{f}_{m,h} + \Delta_{m,1}\hat{f}_{m,h}/\hat{f}_{m,1}$.

In general, a suitable number of frequency estimation iterations depends on the signal. As Fig. 6.8 shows, increasing the number of iterations can increase the quality monotonically when only one source is present. However, when multiple sounds are present, nearby harmonics of other sources can disturb the fundamental frequency estimation of a source. In this case a large number of frequency estimation iterations can cause the fundamental frequency of a source to become adjusted to a wrong source. Therefore, we found it useful not to allow the fundamental frequencies to change more than 10% from the original multiple fundamental frequency estimates. The effect of frequency updating on the average SDR was found to be rather small, as the simulation experiments in Section 6.6 show.

Estimating the frequencies of partials which cross each other in the time-frequency place is a difficult task which can be solved only in limited cases using special methods, for example as proposed in [41]. Since here the overtone frequencies are tied to the fundamental frequency, the method is able to solve crossing partials (see Fig. 6.9), as long as the fundamental frequencies do not cross each other.

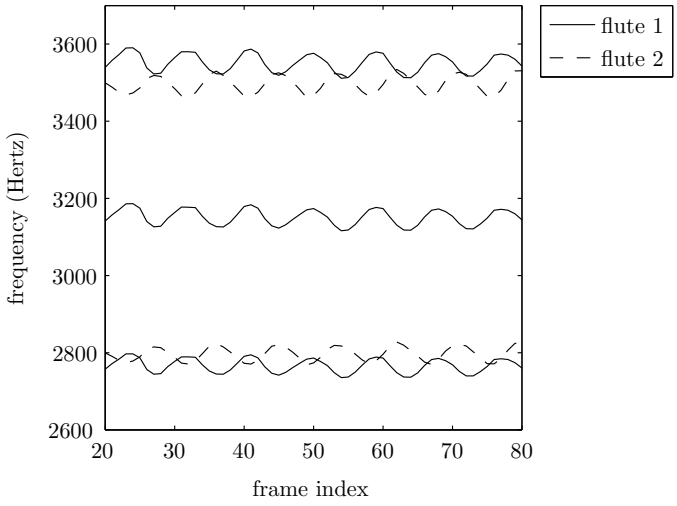

Figure 6.9: Estimated overtone frequencies of a mixture signal consisting of two simultaneous flutes with vibrato. The frequency estimation uses all the overtones of the sounds, and is therefore able to solve also crossing partials in some cases.

6.5 Combining Separated Sinusoids into Notes

The previous sections explained the parameter estimation in a single MFFE frame. Since onsets of other instruments can occur during a musical note, a note can be divided into several adjacent MFFE frames. Our target is to combine the signals separated within each MFFE frame to get entire musical notes.

Since the fundamental frequency variation within a note is often relatively small, a simple approach is to combine closely spaced fundamental frequency values in adjacent frames into notes. Sharp increases in the overtone amplitudes can be interpreted as new note onsets.

More robust results were obtained by using the method proposed by Ryynänen and Klapuri [149], which estimates the notes by modeling the MFFE features using hidden Markov models. The algorithm estimates a set of notes, each of which has a fundamental frequency, onset time, and offset time. Instead of the raw fundamental frequency estimates, we use the fundamental frequencies of the entire note events to initialize the nonlinear least-squares algorithm. Since the note estimation algorithm does not produce estimates of overtones frequencies, we have to assume perfect harmonicity.

To enable flexible testing of the algorithms we implemented a pianoroll-type graphical interface, where each note is drawn with a filled patch (see Fig. 6.10). The location on the x-axis indicates the note timing, and the location on the y-axis shows its fundamental frequency. Variance in the vertical location is used to illustrate time-varying fundamental frequency, and the thickness of each note

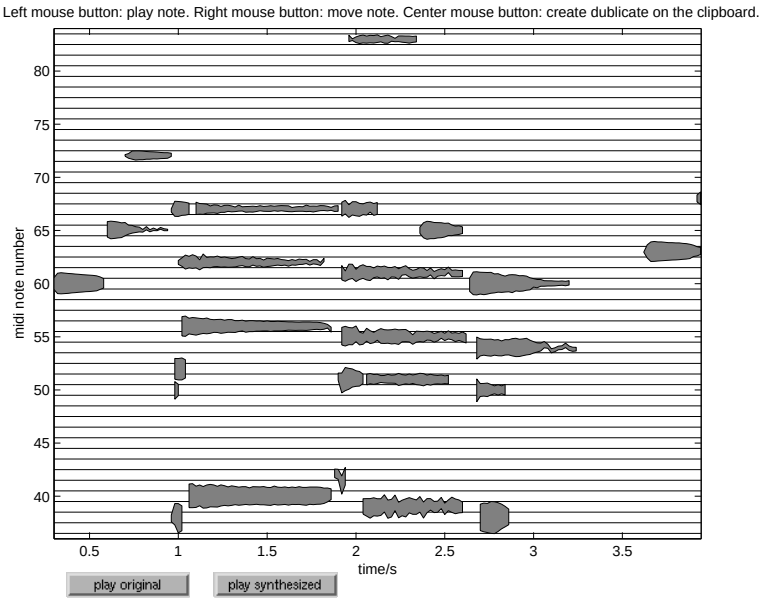

Figure 6.10: A pianoroll-type representation of the notes separated from an excerpt of polyphonic music (Piece 18 from the RWC Jazz Music Database [69]).

illustrates the approximated loudness of the note. The interface allows editing the signal by moving individual notes in time and pitch, and deleting, and copying notes. A Matlab implementation of the interface and example signals are available at `http://www.cs.tut.fi/~tuomasv/`.

The separated notes can be further grouped into sound sources. Since the separation performance of existing algorithms is quite limited, there are no studies where the separation and grouping was done completely automatically using only acoustic signals. Every [51] used pattern recognition and clustering approach on a material where the separation was guided by a MIDI reference. He tested numerous features and model-based clustering, which models the probability of a note belonging to a source by a normal distribution centered on a source cluster center.

6.6 Simulation Experiments

Since the main scientific contribution in this chapter is the parameter estimation of overlapping sources, the simulation concentrate only on the proposed nonlinear least-squares algorithm within one MFFE frame.

6.6.1 Acoustic Material

The performance of the proposed methods was evaluated using generated mixtures of harmonic sounds. Test material consisted of a database of 26 different musical instruments comprising plucked and bowed string instruments, flutes, brass, and reed instruments. These introduce several different sound production mechanisms and a variety of spectra. The database was a combination of samples from the McGill University Master Samples Collection [131], the University of Iowa website [89], and samples recorded from the Roland XP-30 synthesizer. The total number of samples available for generating the mixtures was 4128.

Random sound mixtures of one to six simultaneous samples were generated by selecting one to six samples randomly from the database. Each sample was selected by first allotting an instrument and then a random note from its whole playing range, however, restricting the fundamental frequency over five octaves between 65 Hz and 2100 Hz. To model the dynamic differences between sounds in real recordings, each sample was scaled to have a random power between 0 and 10 dB. The selected samples were summed to obtain a mixture signal. For each polyphony we generated 100 mixtures.

6.6.2 Algorithms

The discussed separation algorithms enable a large number of variations and different implementations. All the methods tested here were based on the same implementation and the explained iterative parameter estimation procedure. The main differences between the algorithms are in the estimation of overlapping partials. Some parameter values such as the window size were varied to study their effect. Unless otherwise stated, a single frequency estimation iteration and a 40 ms window with 50% overlap between the frames was used, and overtones were considered to be colliding if they were not more than $0.5 f_s/N$ Hertz apart from each other.

The evaluated algorithms were the following:

- Comb filtering. In this method the amplitude of a sinusoid is determined directly from the amplitude spectrum of the mixture at the partial frequency. Thus, the method does not include a mechanism for resolving overlapping harmonics. The implementation of the method is based on the sinusoidal model.

- Linear interpolation and normalization. Linear interpolation according to (5.3) is the most commonly used method, and here it is followed by the normalization (5.4) since it provides a small improvement.

- Nonlinear smoothing. This applies the nonlinear filter of Klapuri [105] on the amplitude spectrum of each sound to reduce the effect of overlapping components, as done in [194].

- Fixed frequency-band model proposed on Page 80. For source m, the center frequency of the l^{th} band is set equal to the frequency of the $\lceil 1.5^{l-1} \rceil^{\text{th}}$

overtone.

- Adaptive frequency-band model proposed on Page 83.

The results for methods using linear interpolation without normalization, the polynomial model, and the Mel-frequency cosine basis model are presented in Appendix B.

6.6.3 Evaluation of the Separation Quality

Acoustic input was fed to the separation algorithm and the frequencies, amplitudes and phases of each sound were estimated. Separated sounds were synthesized by interpolating the parameters from frame to frame, as in [120]. Each separated signal was matched to an original signal based on its fundamental frequency. When the MFFE estimated the number of sounds and their fundamental frequencies correctly, there was a one-to-one match between the separated signals and the original ones. In the case of erroneous fundamental frequency estimates, for each original signal we sought for a separated signal which has an equal fundamental frequency, and matched them. Remaining separated signals were discarded, and original signals which did not have a separated signal of equal fundamental frequency were set to match to a separated signal of all zeros.

Since the separation algorithms preserve the phase of the signal, an error between the original and synthesized signal could be obtained simply by subtracting the synthesized signal from the original ones in the time domain.

The mean-square level of the error signal was computed over the whole signal and compared to the original to obtain signal-to-distortion ratio (SDR, (1.2)) of the separated signal. The separation system limits the frequencies of the sinusoids below 5kHz, thus the comparisons were limited to the frequency band 0-5kHz by low-pass filtering the signals. To enable more accurate measurement of the harmonic segments, we did not use the first 30 ms segments of the signals, which were assumed to contain nonharmonic attack transients.

The SDRs in dB were averaged over all the signals. Since the accuracy of the MFFE affects the separation performance significantly, we considered separately cases where the MFFE estimates were correct and erroneous. A mixture was regarded as erroneous, if the MFFE estimated at least one fundamental frequency in the mixture incorrectly. The error rates for the multiple pitch estimator for mixtures ranging from one to six simultaneous sounds were 4.8%, 48.5%, 66.1%, 86.4%, 93.8%, and 100%, respectively. The accuracies are worse than those in [105] because of level differences between the samples.

Also the segmental SDRs, were calculated. Before averaging the frame-wise SDRs were limited between 0 and 30 dB, since in a few frames the original signal or the residual can be vanishing, which would result in very large positive or negative SDRs, thus dominating the average.

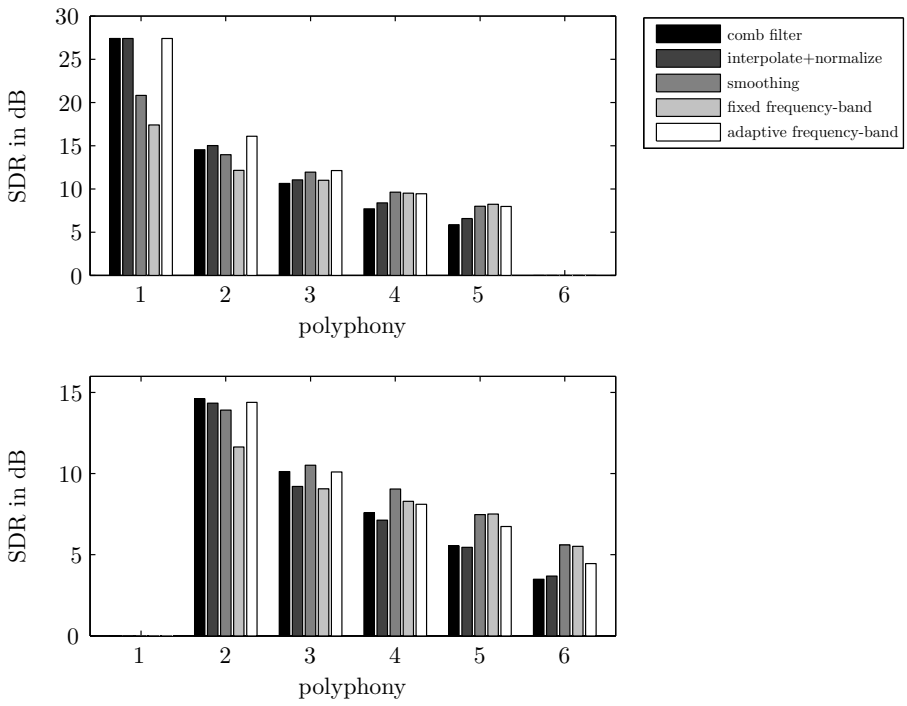

Figure 6.11: The average signal-to-distortion ratios of different algorithms in different polyphonies in the case of correct (upper plot) and erroneous (lower plot) MFFE.

6.6.4 Results

Figure 6.11 shows the average SDRs for each algorithm at each polyphony in the case of correct and erroneous MFFE. The most significant tendency in the results is the gradual reduction in quality when the polyphony increases. Compared with the effect of the polyphony, the average differences between algorithms are small, except in the polyphony one, which shows the modeling error caused by the sinusoidal representation. The best methods achieve SDRs of about 30 dB, which can be considered to be sufficient for most applications. Smoothing and fixed frequency-band model place limitations for the sound spectra, and therefore cause additional modeling error and lower SDR for a single source. However, more strict limitations of smooth spectrum enable better results at higher polyphonies.

At low polyphonies, the best results on average are obtained with the adaptive frequency-band model, whereas on higher polyphonies the best results are obtained with the fixed frequency band model and smoothing. The results for correct MFFE are on average 0-2 dB better that the results with erroneous MFFE. The difference is surprisingly small, but this is partly because a mix-

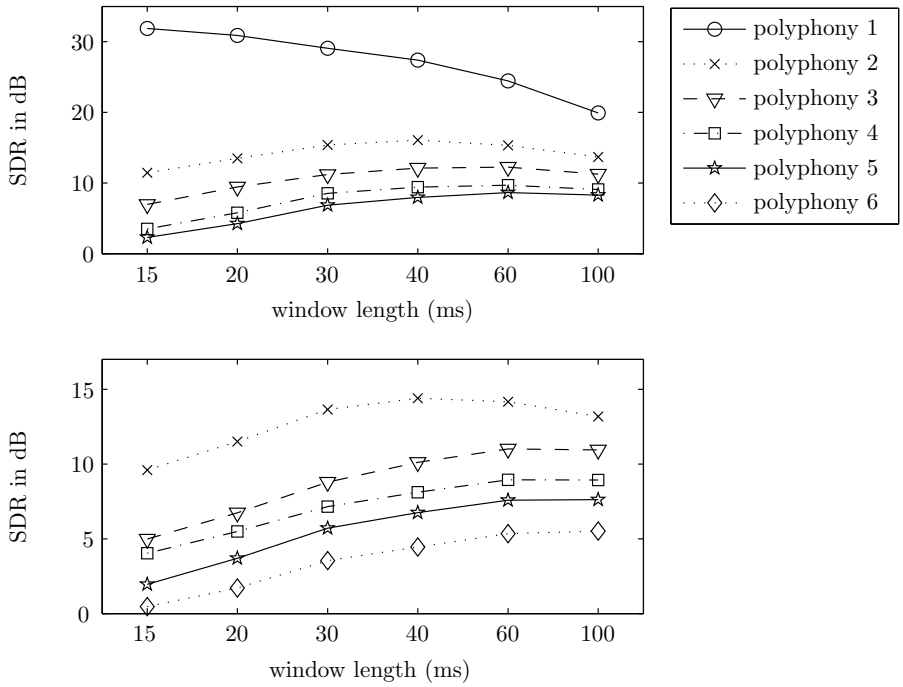

Figure 6.12: The signal-to-distortion ratios for adaptive frequency-band model at different window sizes for correct (upper panel) and erroneous (lower panel) MFFE.

ture was considered erroneous even when a single fundamental frequency was incorrectly estimated.

Figure 6.12 shows the results of the adaptive frequency-band model for different window sizes. For a single source, smaller window lengths enable a smaller modeling error. This is natural since a small window can enable more accurate time-varying parameters. For multiple sources, the use of a too short or too long window reduces the quality. In the case of a short window, the resolution of the spectrum is not sufficient for resolving closely spaced overtones. In the case of a long window, the temporal resolution becomes worse. A good selection of the window length depends on the polyphony and accuracy of the MFFE. For correct MFFE and more than one source, approximately 40 ms window produced the best results. For erroneous MFFE a slightly longer, approximately 60 ms window produced the best results.

Figure 6.13 illustrates the performance of the adaptive frequency-band model for different numbers of iterations. For a single source the quality increases monotonically as the number of iterations increases. This is natural since the estimation algorithm obtains a better fit for the model at each iteration, and overlapping sources do not affect the procedure. For multiple sources the average

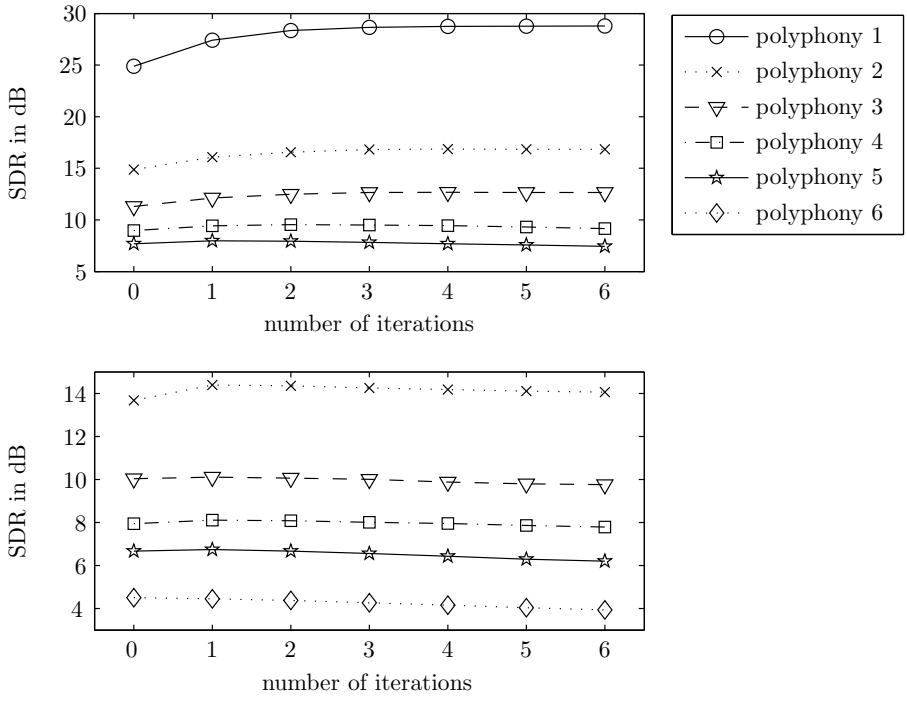

Figure 6.13: The signal-to-distortion ratios for adaptive frequency-band model with different number of frequency estimation iterations for correct (upper panel) and erroneous (lower panel) MFFE.

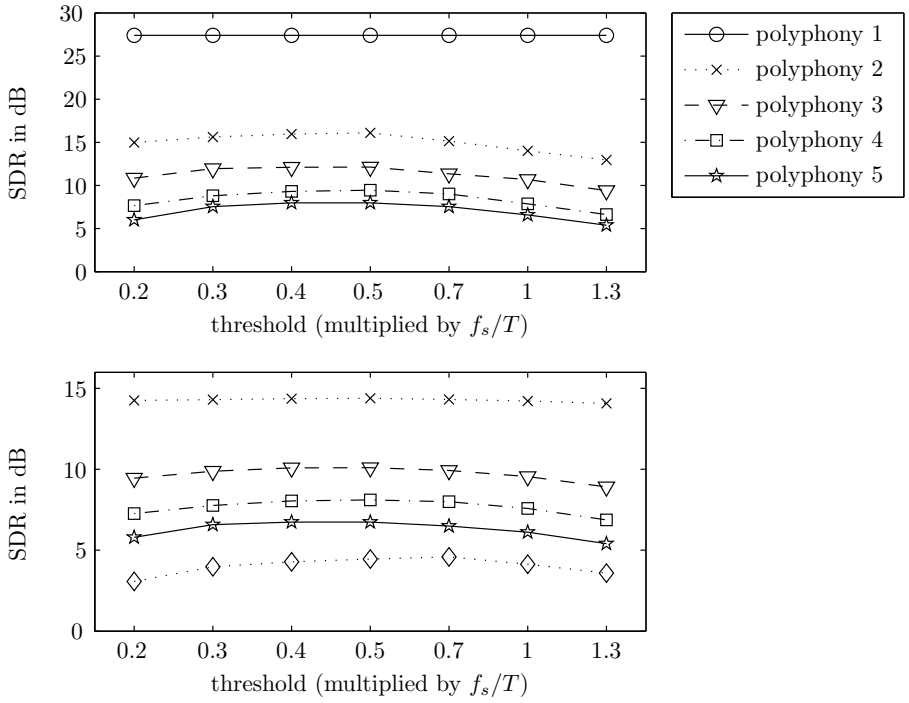

Figure 6.14: The average SDRs for adaptive frequency-band model with different frequency thresholds for determining colliding overtones. The upper panel shows the results for correct MFFE and the lower panel for erroneous MFFE.

effect of frequency estimation iterations is small. For correct MFFE a small improvement is obtained, whereas for erroneous MFFE the quality is slightly decreased.

Figure 6.14 shows the average SDRs when the frequency threshold for determining colliding overtones was varied. In the case of a single sound the threshold does not affect the performance, since there are no overlapping or nearby overtones. In the case of multiple sources, the threshold of $0.5f_s/N$ produces good results on the average. The parameters of closely spaced partials can be estimated as long as the frequency difference is large enough. When too closely spaced partials are tried to estimate, this may cause large estimation errors. On the other hand, interpolating partials is always likely to cause some errors.

Figure 6.15 shows the measured segmental SDRs for the same task as those shown in Figure 6.11. On the average, the results are approximately equal to the normal SDRs presented in Figure 6.11. For polyphony one the segmental SDRs are slightly lower, since the frame-wise SDRs were limited below 30 dB before averaging. For higher polyphonies the segmental SDRs are higher than normal SDRs. Larger errors are more likely in frames where the energies are

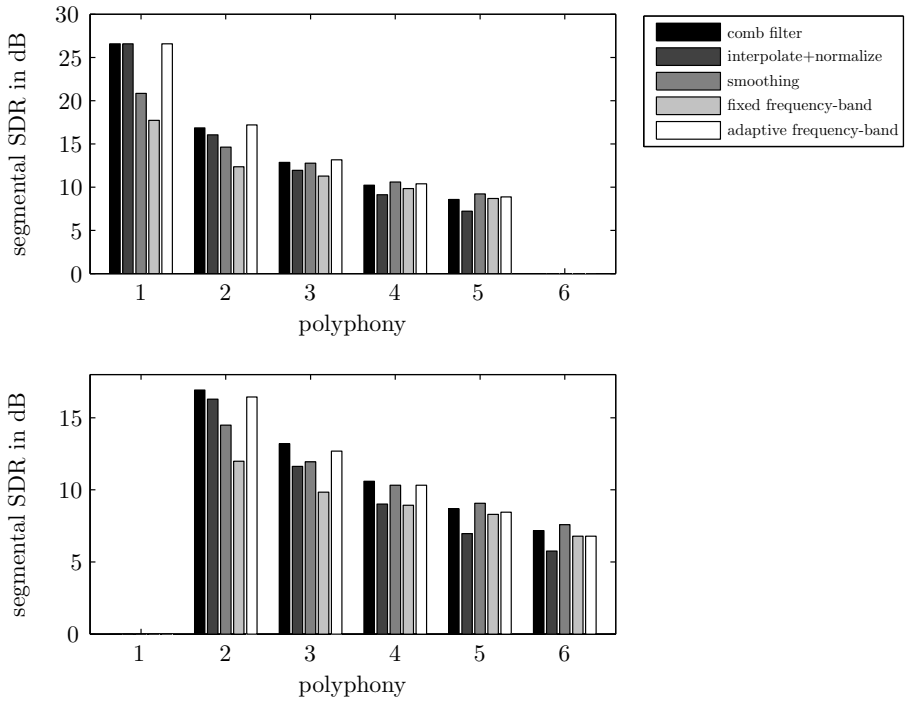

Figure 6.15: The average segmental signal-to-distortion ratios for different algorithms in different polyphonies in the case of correct (upper panel) and erroneous (lower panel) MFFE.

large, which results in a higher average segmental SDR. The most significant difference in the segmental SDR to the normal SDR between different algorithms is the good performance of comb filtering. Even though the comb filtering does not have a mechanism for resolving overlapping harmonics, other methods did not produce significantly better average segmental SDRs. The reason for this is partly in the evaluation procedure: because the comb filter sets the amplitude of a sinusoid equal to the amplitude of the mixture at the sinusoid's frequency, it is likely to produce a large relative errors in frames where a source has a small energy compared with other sources. Since the frame-wise SDRs were limited to between 0 and 30 before averaging these frames do not have as significant effect on the segmental SDR as on the normal SDR.

Chapter 7

Conclusions

Blind separation of one-channel signals from a polyphonic mixture is an underdetermined problem, since there are more unknown variables to be estimated than there are known mixture samples. However, signals produced by music instruments have properties which make the separation feasible. The properties utilized in this thesis are redundancy and repetition in time, sparseness of their time-frequency representations, continuity in time and frequency, and harmonic spectral structure. These properties, in addition to other prior knowledge, enable estimating source signals which are close to the original ones before mixing. Much of the engineering work in the development of one-channel sound source separation algorithms is seeking for efficient algorithms which can utilize these properties.

7.1 Unsupervised Learning Algorithms

In Chapter 3, we proposed an unsupervised learning algorithm for monaural sound source separation which is based on non-negative matrix factorization of the magnitude spectrogram with temporal continuity and sparseness objectives. Since many musical sounds can be approximated as a sum of components which have a fixed spectrum and time-varying gain, relatively simple and efficient algorithms like PCA can be used to estimate the sources in the unsupervised learning framework. However, only relative new algorithms such as ICA and NMF enable a quality which is sufficient for practical applications. In many separation and analysis problems the best results are currently obtained with the non-negativity restrictions, originally proposed for this framework simultaneously by Smaragdis and Brown in [166] and Virtanen in [189].

Most of the existing algorithms that are based on the linear instantaneous model for a spectrogram are limited in a sense that they consider each frame as an individual observation, even though natural sounds are often slowly-varying in time. The proposed cost function which is the sum of the squared differences between the gains of adjacent frames is a simple and efficient way of adding

the temporal continuity objective to this separation framework. The simulation experiments show that the temporal continuity criterion improves the detection accuracy and SDRs of pitched instrument sounds. The sparseness assumptions, in turn, did not lead to significantly better results. The non-negative matrix factorization algorithms were shown to produce significantly better separation results than independent component analysis.

The optimization algorithm has a significant effect on the separation results. The convergence of the projected steepest descent algorithm is poor in the minimization of the normal divergence, which often produces the best results. Possibilities to overcome this are the proposed multiplicative update rules or the modified divergence measure.

Since most natural sounds have time-varying spectra, approximating them using components with a fixed spectrum sets a limit for the separation quality. In Chapter 4 we proposed two convolutive signal models which extend the instantaneous model to allow either time-varying spectra or frequencies. In addition to our original work in [190], similar extensions were simultaneously published by Smaragdis [164, 165]. Estimation algorithms for the convolutive model were proposed which are based on the minimization of the reconstruction error between the observed magnitude spectrogram and the model while restricting the parameters to non-negative values.

Simulation experiments show that the model which allows time-varying spectra can be used to estimate and represent components which correspond to real-world sound objects. On generated test signals the model was shown to enable higher quality of separated drum signals than the existing algorithms which are based on the instantaneous model. The computational complexity of the proposed algorithm is quite high, limiting the length of target signals, so that improvements in the separation of longer entities such as pitched notes could not be obtained.

The model which allows time-varying frequencies was not evaluated systematically; however, the experiments indicate that the model can be used to represent pitched notes with different fundamental frequency values. This enables separation and estimation of harmonic partials which overlap with other sounds by utilizing the spectra of notes with adjacent fundamental frequency values. The model produces representations which are a good basis for the automatic transcription of pitched instruments in music.

Compared to other approaches towards monaural sound source separation, the proposed unsupervised learning methods enable a relatively good separation quality – although it should be noted that the performance in general is still very limited. A strength of the presented methods is their scalability: the methods can be used for arbitrarily complex material. In the case of simple monophonic signals, they can be used to separate individual notes, and in complex polyphonic material, the algorithms can extract larger repeating entities, such as chords.

As discussed in Section 1.5, virtually all evaluations of unsupervised or blind source separation systems can be claimed to be somewhat limited. This thesis is not an exception: even though we try to obtain statistical reliability by using

a large number of test signals which are generated using recorded samples of real-world instruments, we restrict ourselves to rather simple mixing cases, and separated signals are assigned to references by comparing them with the original signals. Since the algorithms proposed in this thesis were not developed for a specific application, we use mainly a single low-level objective measure. Naturally, optimizing an algorithm for a specific measure does not guarantee that the performance improves according to some other measure — especially when the obtained improvements were still relatively small.

7.2 Sinusoidal Modeling

The sinusoidal model discussed in Chapter 5 provides a good basis for the separation of pitched sounds. Each harmonic partial can be represented with a single sinusoid, and they can be robustly estimated by minimizing the error between the model and the observed signal, while restricting the overtones to harmonic frequency relationships. Because the rough spectral shape of natural sounds is continuous as a function of frequency, the amplitudes of overlapping partials can be approximated by interpolation, for which we proposed and tested numerous interpolation methods in Chapter 6.

To estimate the sinusoids we proposed a computationally efficient nonlinear least-squares algorithm which first initialized the frequencies with a multiple fundamental frequency estimator, and then iteratively estimates the amplitudes and phases and improves the frequency estimates. For the frequency estimation we proposed a method which linearizes the error with respect to fundamental frequencies and enables updating the fundamental frequency estimates efficiently while retaining the harmonic spectral structure.

The simulation experiments show that the quality of the separation is good for low polyphonies, and it decreases gradually as the number of simultaneous sounds increases. Approximating overlapping partials with any tested interpolation method produces reasonable results and none of the tested methods performed clearly better than the others.

7.3 Discussion and Future Work

The most significant tendency in monaural separation algorithms is the development of unsupervised learning methods: about 5-10 years ago their performance was not sufficient for separating real-world acoustic signals. Increased computational power, storage capacity, and development of machine learning algorithms have enabled approaches where the characteristics of signals are learned from data. This reduces the need for incorporating information about the data into the algorithms manually.

Despite the proposed and discussed improvements, monaural sound source separation is still largely an unsolved problem, and there are some clear shortcomings in the existing algorithms. For example, a challenge with the presented

unsupervised learning based separation algorithms is that it is difficult to restrict the sources to be harmonic: the existing methods can implement the restriction only by fixing the fundamental frequency of a component and allowing non-zero energy only at predicted overtone frequencies. Other future work includes development of methods for estimating the number of components and automatic clustering. Future work in the separation algorithms based on sinusoidal modeling includes the use of temporal continuity and other notes of the same instrument in more accurate estimation of overlapping partials.

Appendix A

Convergence Proofs of the Update Rules

A.1 Augmented Divergence

In this section we show that the augmented divergence

$$D_\epsilon(\mathbf{X}||\mathbf{BG}) = \sum_{k,t}([\mathbf{X}]_{k,t}+\epsilon)\log\frac{[\mathbf{X}]_{k,t}+\epsilon}{[\mathbf{BG}]_{k,t}+\epsilon} - [\mathbf{X}]_{k,t} + [\mathbf{BG}]_{k,t} \tag{A.1}$$

is non-increasing under the update rule

$$\mathbf{B} \leftarrow \mathbf{B}.\times\frac{\frac{\mathbf{X}+\epsilon}{\mathbf{BG}+\epsilon}\mathbf{G}^T}{\mathbf{1}\mathbf{G}^T}. \tag{A.2}$$

Let us write the augmented divergence as a function of $\mathbf{B}$ as

$$F = D_\epsilon(\mathbf{X}||\mathbf{BG}) \tag{A.3}$$

The proof is based on forming an *auxiliary function* [114] for F, given as

$$\begin{aligned} G(\mathbf{B},\mathbf{B}') = &\sum_{k,t}([\mathbf{X}]_{k,t}+\epsilon)\log([\mathbf{X}]_{k,t}+\epsilon) - [\mathbf{X}]_{k,t} + [\mathbf{BG}]_{k,t} \\ &- ([\mathbf{X}]_{k,t}+\epsilon)\sum_{j=1}^{J}\frac{[\mathbf{G}]_{j,t}[\mathbf{B}']_{k,j}}{[\mathbf{B}'\mathbf{G}]_{k,t}+\epsilon}\left(\log([\mathbf{G}]_{j,t}[\mathbf{B}]_{k,j}) - \log\frac{[\mathbf{G}]_{j,t}[\mathbf{B}']_{k,j}}{[\mathbf{B}'\mathbf{G}]_{k,t}+\epsilon}\right) \\ &- \epsilon([\mathbf{X}]_{k,t}+\epsilon)\frac{\log([\mathbf{B}'\mathbf{G}]_{k,t}+\epsilon)}{[\mathbf{B}'\mathbf{G}]_{k,t}+\epsilon}, \end{aligned}$$

where the argument $\mathbf{B}'$ is a non-negative matrix of the same size as $\mathbf{B}$. An auxiliary function fulfills two properties: $G(\mathbf{B},\mathbf{B}) = F(\mathbf{B})$ and $G(\mathbf{B},\mathbf{B}') \geq F(\mathbf{B})$. This results in update rule

$$\mathbf{B}'' \leftarrow \arg\min_{\mathbf{B}} G(\mathbf{B},\mathbf{B}'). \tag{A.4}$$

The divergence is non-increasing under update rule (A.4), since $F(\mathbf{B}'') \leq G(\mathbf{B}'', \mathbf{B}') \leq G(\mathbf{B}', \mathbf{B}') = F(\mathbf{B}')$.

First, we prove that G is an auxiliary function of F. The first condition $G(\mathbf{B}, \mathbf{B}) = F(\mathbf{B})$ is straightforward to verify. The second condition $G(\mathbf{B}, \mathbf{B}') \geq F(\mathbf{B})$ is verified as follows. First, logarithm is a concave function, so that by Jensen's inequality [148, pp. 62-63] we can write

$$\sum_{j=0}^{J} \alpha_j \log(r_j) \leq \log(\sum_{j=0}^{J} \alpha_j r_j) \tag{A.5}$$

for any non-negative α_j for which $\sum_{j=0}^{J} \alpha_j = 1$. Let us set $y_j = r_j \alpha_j$, so that the above equals

$$\sum_{j=0}^{J} \alpha_j \log(y_j/\alpha_j) \leq \log(\sum_{j=0}^{J} y_j). \tag{A.6}$$

Let us set $y_0 = \epsilon$, so that we get

$$-\log(\epsilon + \sum_{j=1}^{J} y_j) \leq -\sum_{j=1}^{J} \alpha_j \log(y_j/\alpha_j) - \alpha_0 \log(\epsilon/\alpha_0) \tag{A.7}$$

Let us set $\alpha_j = \frac{[\mathbf{G}]_{j,t}[\mathbf{B}']_{k,j}}{[\mathbf{B}'\mathbf{G}]_{k,t}+\epsilon}$, $y_j = [\mathbf{G}]_{j,t}[\mathbf{B}]_{k,j}$ for $j = 1, \ldots, J$, and $\alpha_0 = \frac{\epsilon}{[\mathbf{B}'\mathbf{G}]_{k,t}+\epsilon}_{k,t}$, for any k and t, so that we get

$$\begin{aligned} -\log(\epsilon + [\mathbf{B}\mathbf{G}]_{k,t}) \leq & -\sum_{j=1}^{J} \frac{[\mathbf{G}]_{j,t}[\mathbf{B}']_{k,j}}{[\mathbf{B}'\mathbf{G}]_{k,t}+\epsilon} \left[\log([\mathbf{G}]_{j,t}[\mathbf{B}]_{k,j}) - \log(\frac{[\mathbf{G}]_{j,t}[\mathbf{B}']_{k,j}}{[\mathbf{B}'\mathbf{G}]_{k,t}+\epsilon}) \right] \\ & - \frac{\epsilon}{[\mathbf{B}'\mathbf{G}]_{k,t}+\epsilon} \log(\epsilon + [\mathbf{B}'\mathbf{G}]_{k,t}). \end{aligned}$$

By multiplying both sides by $([\mathbf{X}]_{k,t} + \epsilon) > 0$ and adding terms $([\mathbf{X}]_{k,t} + \epsilon)\log([\mathbf{X}]_{k,t} + \epsilon) - [\mathbf{X}]_{k,t} + [\mathbf{B}\mathbf{G}]_{k,t}$, we get

$$\begin{aligned} & ([\mathbf{X}]_{k,t} + \epsilon) \log(\frac{[\mathbf{X}]_{k,t} + \epsilon}{[\mathbf{B}\mathbf{G}]_{k,t} + \epsilon}) - [\mathbf{X}]_{k,t} + [\mathbf{B}\mathbf{G}]_{k,t} \leq \\ & - ([\mathbf{X}]_{k,t} + \epsilon) \sum_{j=1}^{J} \frac{[\mathbf{G}]_{j,t}[\mathbf{B}']_{k,j}}{[\mathbf{B}'\mathbf{G}]_{k,t}+\epsilon} \left[\log([\mathbf{G}]_{j,t}[\mathbf{B}]_{k,j}) - \log(\frac{[\mathbf{G}]_{j,t}[\mathbf{B}']_{k,j}}{[\mathbf{B}'\mathbf{G}]_{k,t}+\epsilon}) \right] \\ & - ([\mathbf{X}]_{k,t} + \epsilon) \left[\frac{\epsilon \log([\mathbf{B}'\mathbf{G}]_{k,t} + \epsilon)}{[\mathbf{B}'\mathbf{G}]_{k,t}+\epsilon} - \log([\mathbf{X}]_{k,t} + \epsilon) \right] - [\mathbf{X}]_{k,t} + [\mathbf{B}\mathbf{G}]_{k,t} \end{aligned}$$

The above holds for all k, t. By summing both sides over k, t, the left-hand side equals $F(\mathbf{B})$, and the right-hand side $G(\mathbf{B}, \mathbf{B}')$. Therefore, $F(\mathbf{B}) \leq G(\mathbf{B}, \mathbf{B}')$, and G is an auxiliary function of F.

Let us solve (A.4) by setting the gradient of G with respect to $\mathbf{B}$ equal to zero. The partial derivative of G with respect to $\mathbf{B}_{k',j'}$ is given as

$$\frac{\nabla G_\epsilon(\mathbf{B},\mathbf{B}')}{\nabla \mathbf{B}_{k',j'}} = \sum_{k,t}[\mathbf{G}]_{j,t} - ([\mathbf{X}]_{k,t}+\epsilon)\frac{[\mathbf{G}]_{j,t}[\mathbf{B}']_{k',j'}}{[\mathbf{B}'\mathbf{G}]_{k,t}+\epsilon}\cdot\frac{1}{\mathbf{B}_{k',j'}} = 0, \tag{A.8}$$

from which we solve $[\mathbf{B}]_{k',j'}$ as

$$[\mathbf{B}]_{k',j'} = [\mathbf{B}']_{k',j'}\cdot\frac{\sum_t[\mathbf{G}]_{j,t}\frac{[\mathbf{X}]_{k,t}+\epsilon}{[\mathbf{B}'\mathbf{G}]_{k,t}+\epsilon}}{\sum_t[\mathbf{G}]_{j,t}}. \tag{A.9}$$

For each k', j' this equals the update rule (A.2). Since the above is solution to (A.4), under which the augmented divergence which was shown to be non-increasing, the divergence is non-increasing under update rule (A.2).

A.2 Convolutive Model

This section proves that the Euclidean distance (2.18) and divergence (2.19) between the observation matrix $\mathbf{X}$ and model

$$\hat{\mathbf{X}} = \sum_{\tau=0}^{L-1}\mathbf{B}_\tau\cdot\underset{\tau\rightarrow}{\mathbf{G}}, \tag{A.10}$$

where the operator $\underset{\tau\rightarrow}{(\cdot)}$ is defined as in (4.3), is non-increasing under update rules (4.10) - (4.13).

Lee and Seung [114] showed that for non-negative matrices the Euclidean distance and divergence between observation matrix $\mathbf{X}$ and model $\hat{\mathbf{X}} = \mathbf{BG}$ is non-increasing under update rules (2.21) - (2.24). The update rules (4.10) - (4.13) can be derived from these update rules.

A.2.1 Event Spectrograms

The convolutive signal model (A.10) can be written in the form $\hat{\mathbf{X}} = \mathbf{B}\mathbf{G}'$ by using

$$\mathbf{B} = \begin{bmatrix}\mathbf{B}_0\dots\mathbf{B}_{L-1}\end{bmatrix} \tag{A.11}$$

and

$$\mathbf{G}' = \begin{bmatrix}\mathbf{G}^\mathsf{T} & \underset{1\rightarrow}{\mathbf{G}}^\mathsf{T}\cdots\underset{(L-1)\rightarrow}{\mathbf{G}}^\mathsf{T}\end{bmatrix}^\mathsf{T} \tag{A.12}$$

By substituting the above into (2.21), we get

$$\begin{bmatrix}\mathbf{B}_0\dots\mathbf{B}_{L-1}\end{bmatrix} \leftarrow \begin{bmatrix}\mathbf{B}_0\dots\mathbf{B}_{L-1}\end{bmatrix} .\times \frac{\mathbf{X}\cdot\begin{bmatrix}\mathbf{G}^\mathsf{T} & \underset{1\rightarrow}{\mathbf{G}}^\mathsf{T}\cdots\underset{(L-1)\rightarrow}{\mathbf{G}}^\mathsf{T}\end{bmatrix}}{\hat{\mathbf{X}}\cdot\begin{bmatrix}\mathbf{G}^\mathsf{T} & \underset{1\rightarrow}{\mathbf{G}}^\mathsf{T}\cdots\underset{(L-1)\rightarrow}{\mathbf{G}}^\mathsf{T}\end{bmatrix}}, \tag{A.13}$$

and by substituting them into (2.23), we get

$$\left[\mathbf{B}_0 \dots \mathbf{B}_{L-1}\right] \leftarrow \left[\mathbf{B}_0 \dots \mathbf{B}_{L-1}\right] .\times \frac{\left[\frac{\mathbf{X}}{\hat{\mathbf{X}}}\right] \cdot \left[\mathbf{G}^{\mathsf{T}} \quad \underset{1\rightarrow}{\mathbf{G}}^{\mathsf{T}} \dots \underset{(L-1)\rightarrow}{\mathbf{G}}^{\mathsf{T}}\right]}{\mathbf{1} \cdot \left[\mathbf{G}^{\mathsf{T}} \quad \underset{1\rightarrow}{\mathbf{G}}^{\mathsf{T}} \dots \underset{(L-1)\rightarrow}{\mathbf{G}}^{\mathsf{T}}\right]}. \tag{A.14}$$

For each $\mathbf{B}_\tau$, these equal the update rules given in Equations (4.11) and (4.13). Therefore, the Euclidean distance is non-increasing under update rule (4.11) and the divergence is non-increasing under update rule (4.13).

A.2.2 Time-Varying Gains

The convolutive signal model (A.10) can be written in a linear form $\hat{\mathbf{x}} = \mathcal{B}\mathbf{g}$ by representing matrices $\mathbf{X}$ and $\hat{\mathbf{X}}$ as vectors

$$\mathbf{x} = \begin{bmatrix} \mathbf{x}_1 \\ \vdots \\ \mathbf{x}_T \end{bmatrix} \quad \hat{\mathbf{x}} = \begin{bmatrix} \hat{\mathbf{x}}_1 \\ \vdots \\ \hat{\mathbf{x}}_T \end{bmatrix}. \tag{A.15}$$

The time-varying gains are represented using

$$\mathbf{g} = \begin{bmatrix} \mathbf{g}_1 \\ \vdots \\ \mathbf{g}_T \end{bmatrix} \tag{A.16}$$

in which $\mathbf{g}_t = \left[g_{1,t} \dots g_{J,t}\right]^{\mathsf{T}}$, and matrix $\mathcal{B}$ is defined to be

$$\mathcal{B} = \begin{bmatrix}
\mathbf{B}_0 & \mathbf{B}_1 & \cdots & \mathbf{B}_{L-1} & \mathbf{0} & \cdots & & \mathbf{0} & & \cdots & \mathbf{0} \\
\mathbf{0} & \mathbf{B}_0 & & \mathbf{B}_{L-2} & \mathbf{B}_{L-1} & & & \vdots & & & \vdots \\
\vdots & & \ddots & & \vdots & & & & & & \\
\mathbf{0} & & & \mathbf{B}_0 & \mathbf{B}_1 & & & \mathbf{0} & & & \mathbf{0} \\
 & & & \mathbf{0} & \mathbf{B}_0 & & & & & & \\
\vdots & & & & & \ddots & & \vdots & & & \vdots \\
 & & & & & & \mathbf{B}_0 & \mathbf{B}_1 & & \mathbf{B}_{L-1} & \mathbf{0} \\
\mathbf{0} & & & & & & \mathbf{0} & \mathbf{B}_0 & & \mathbf{B}_{L-2} & \mathbf{B}_{L-1} \\
 & & & & & & & & \ddots & & \vdots \\
\vdots & & & & & & & & & \mathbf{B}_0 & \mathbf{B}_1 \\
\mathbf{0} & & \cdots & & \mathbf{0} & \cdots & & \mathbf{0} & \cdots & \mathbf{0} & \mathbf{B}_0
\end{bmatrix} \tag{A.17}$$

in which $\mathbf{0}$ is all-zero matrix of the same size as $\mathbf{B}_\tau$.

The update rules (2.22) and (2.24) for the model $\hat{\mathbf{x}} = \mathcal{B}\mathbf{g}$ can be written as

$$\mathbf{g} \leftarrow \mathbf{g} .\times \frac{\mathcal{B}^{\mathsf{T}}\mathbf{x}}{\mathcal{B}^{\mathsf{T}}\hat{\mathbf{x}}}, \tag{A.18}$$

and

$$\mathbf{g} \leftarrow \mathbf{g}.\times \frac{\mathcal{B}^{\mathsf{T}}[\frac{\mathbf{x}}{\hat{\mathbf{x}}}]}{\mathcal{B}^{\mathsf{T}}\underline{1}}, \tag{A.19}$$

where $\underline{1}$ is an all-one vector of the same size as $\mathbf{x}$.

For clarity, we write $\mathcal{B}^{\mathsf{T}}$ as

$$\mathcal{B}^{\mathsf{T}} = \begin{bmatrix}
\mathbf{B}_0^{\mathsf{T}} & \mathbf{0}^{\mathsf{T}} & \cdots & \mathbf{0}^{\mathsf{T}} & & \cdots & & \mathbf{0}^{\mathsf{T}} & & \cdots & \mathbf{0}^{\mathsf{T}} \\
\mathbf{B}_1 & \mathbf{B}_0^{\mathsf{T}} & & & & & & & & & \\
\vdots & & \ddots & & & & & & & & \vdots \\
\mathbf{B}_{L-1}^{\mathsf{T}} & \mathbf{B}_{L-2}^{\mathsf{T}} & & \mathbf{B}_0^{\mathsf{T}} & \mathbf{0}^{\mathsf{T}} & & & & & & \\
\mathbf{0}^{\mathsf{T}} & \mathbf{B}_{L-1}^{\mathsf{T}} & \cdots & \mathbf{B}_1^{\mathsf{T}} & \mathbf{B}_0^{\mathsf{T}} & & & & & & \mathbf{0}^{\mathsf{T}} \\
\vdots & & & & & \ddots & & & & & \vdots \\
 & & & & & & \mathbf{B}_0^{\mathsf{T}} & \mathbf{0}^{\mathsf{T}} & & & \\
\mathbf{0}^{\mathsf{T}} & \cdots & & \mathbf{0}^{\mathsf{T}} & & \cdots & \mathbf{B}_1^{\mathsf{T}} & \mathbf{B}_0^{\mathsf{T}} & & & \mathbf{0}^{\mathsf{T}} \\
 & & & & & & & & \ddots & & \vdots \\
\cdots & & & & & & \mathbf{B}_{L-1}^{\mathsf{T}} & \mathbf{B}_{L-2}^{\mathsf{T}} & & \mathbf{B}_0^{\mathsf{T}} & \mathbf{0}^{\mathsf{T}} \\
\mathbf{0}^{\mathsf{T}} & \cdots & & \mathbf{0}^{\mathsf{T}} & & \vdots & \mathbf{0}^{\mathsf{T}} & \mathbf{B}_{L-1}^{\mathsf{T}} & \cdots & \mathbf{B}_1^{\mathsf{T}} & \mathbf{B}_0^{\mathsf{T}}
\end{bmatrix} \tag{A.20}$$

By substituting it and (A.15) and (A.16) to Equations (A.18) and (A.19), we get

$$\mathbf{g}_t \leftarrow \mathbf{g}_t.\times \frac{\sum_{\tau=0}^{\max(L-1,T-t)} \mathbf{B}_\tau^{\mathsf{T}} \mathbf{x}_{t+\tau}}{\sum_{\tau=0}^{\max(L-1,T-t)} \mathbf{B}_\tau^{\mathsf{T}} \hat{\mathbf{x}}_{t+\tau}} \tag{A.21}$$

and

$$\mathbf{g}_t \leftarrow \mathbf{g}_t.\times \frac{\sum_{\tau=0}^{\max(L-1,T-t)} \mathbf{B}_\tau^{\mathsf{T}} [\frac{\mathbf{x}_{t+\tau}}{\hat{\mathbf{x}}_{t+\tau}}]}{\sum_{\tau=0}^{\max(L-1,T-t)} \mathbf{B}_\tau^{\mathsf{T}} \underline{\mathbf{1}}} \tag{A.22}$$

for all $t = 1 \ldots T$. For each $\mathbf{g}_t$ these equal the update rules (4.11) and (4.13), thus completing the proof.

Since the update rules for the time-varying frequency model (Section 4.2) can be proved similarly by reformulating the model to a linear model. Because of the dualism of the models, the update rules are actually identical when the variables are changed according to Section 4.3. Therefore, we omit the converge proof of the time-varying frequency model here.

Appendix B

Simulation Results of Sinusoidal Modeling Algorithms

Figure B.1 shows the average SDRs for separation methods based on sinusoidal modeling, which were not included in the results illustrated in Chapter 5. The methods based on interpolation and normalization and linear-frequency band model are included to enable comparison with results in Chapter 5.

The average performance of the interpolation without normalization is in most cases equal or slightly better than the performance of the interpolation with normalization. The performance of the polynomial model and the mel-frequency cosine basis model is approximately equal to the fixed frequency-band model: the models restrict the parameter values, and therefore their modeling error (polyphony 1) is large, but they enable a better quality on higher polyphonies.

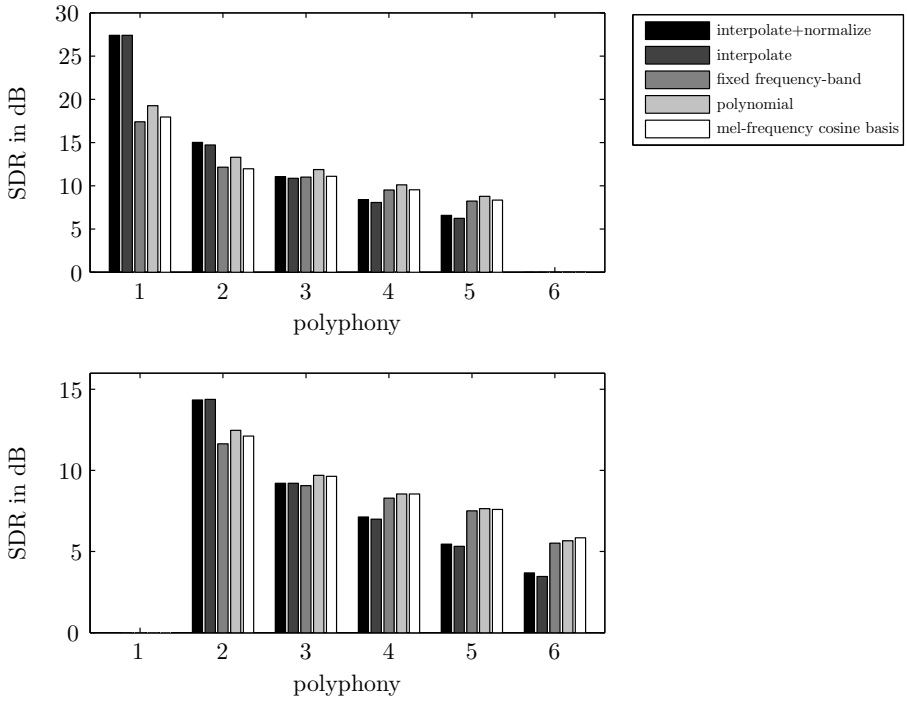

Figure B.1: The average SDRs for the rest sinusoidal modeling algorithms at different polyphonies. The upper panel shows the results for correct MFFE and the lower panel for erroneous MFFE.

Bibliography

[1] S. A. Abdallah. *Towards Music Perception by Redundancy Reduction and Unsupervised Learning in Probabilistic Models*. PhD thesis, Department of Electronic Engineering, King's College London, 2002.

[2] S. A. Abdallah and M. D. Plumbley. If the independent components of natural images are edges, what are the independent components of natural sounds? In *Proceedings of International Conference on Independent Component Analysis and Signal Separation*, pages 534–539, San Diego, USA, 2001.

[3] S. A. Abdallah and M. D. Plumbley. An independent component analysis approach to automatic music transcription. In *Proceedings of the 114th Audio Engineering Society Convention*, Amsterdam, Netherlands, March 2003.

[4] S. A. Abdallah and M. D. Plumbley. Polyphonic transcription by non-negative sparse coding of power spectra. In *Proceedings of International Conference on Music Information Retrieval*, pages 318–325, Barcelona, Spain, October 2004.

[5] M. Abe and S. Ando. Auditory scene analysis based on time-frequency integration of shared FM and AM. In *Proceedings of IEEE International Conference on Audio, Speech and Signal Processing*, Seattle, USA, 1998.

[6] M. Abe and J. O. Smith. Design criteria for simple sinusoidal parameter estimation based on quadratic interpolation of FFT magnitude peaks. In *Proceedings of the 117th Audio Engineering Society Convention*, San Francisco, USA, 2004.

[7] K. Achan, S. Roweis, and B. Frey. Probabilistic inference of speech signals from phaseless spectrograms. In *Proceedings of Neural Information Processing Systems*, pages 1393–1400, Vancouver, Canada, 2003.

[8] X. Amatriain, J. Bonada, A. Loscos, and X. Serra. Spectral processing. In U. Zölzer, editor, *DAFX - Digital Audio Effects*, pages 373–438. John Wiley & Sons, 2002.

[9] C. Andrieu and A. Doucet. Joint Bayesian detection and estimation of noisy sinusoids via reversible jump MCMC. *IEEE Transactions on Signal Processing*, 47(10):2667–2676, 1999.

[10] H. Attias and C. E. Schreiner. Temporal low-order statistics of natural sounds. In *Proceedings of Neural Information Processing Systems*, Denver, USA, 1997.

[11] J. G. Beerends. Audio quality determination based on perceptual measurement techniques. In M. Kahrs and K. Brandenburg, editors, *Applications of Digital Signal Processing to Audio and Acoustics*, pages 1–38. Kluwer Academic Publishers, 1998.

[12] J. G. Beerends, A. P. Hekstra, A. W. Rix, and M. P. Hollier. Perceptual evaluation of speech quality (PESQ), the new ITU standard for end-to-end speech quality assessment, part II — psychoacoustic model. *Journal of the Audio Engineering Society*, 50(10), 2002.

[13] J. G. Beerends and J. A. Stemerdink. A perceptual audio quality measure based on a psychoacoustical sound representation. *Journal of the Audio Engineering Society*, 40(12), 1992.

[14] L. Benaroya, F. Bimbot, and R. Gribonval. Audio source separation with a single sensor. *IEEE Transactions on Audio, Speech, and Language Processing*, 14(1), 2006.

[15] L. Benaroya, F. Bimbot, L. McDonagh, and R. Gribonval. Non negative sparse representation for Wiener based source separation with a single sensor. In *Proceedings of IEEE International Conference on Audio, Speech and Signal Processing*, Hong Kong, China, 2003.

[16] D. P. Bertsekas. *Nonlinear Programming*. Athena Scientific, Belmont, Massachusetts, 1995.

[17] R. B. Blackman and J. W. Tukey. *The Measurement of Power Spectra, From the Point of View of Communications Engineering*. Dover Publications, New York, 1958.

[18] T. Blumensath and M. Davies. Unsupervised learning of sparse and shift-invariant decompositions of polyphonic music. In *Proceedings of IEEE International Conference on Audio, Speech and Signal Processing*, Montreal, Canada, 2004.

[19] T. Blumensath and M. Davies. Sparse and shift-invariant representations of music. *IEEE Transactions on Audio, Speech, and Language Processing*, 14(1), 2006.

[20] R. Bogacz, M. W. Brown, and C. G. Giraud-Carrier. Emergence of movement sensitive neurons' properties by learning a sparse code for natural moving images. In *Proceedings of Neural Information Processing Systems*, pages 838–844, Denver, USA, 2000.

[21] R. N. Bracewell. *The Fourier Transform and Its Applications*. McGraw-Hill, second edition, 1978.

[22] A. Bregman. *Auditory scene analysis*. MIT Press, Cambridge, USA, 1990.

[23] J. C. Brown. Musical fundamental frequency tracking using a pattern recognition method. *Journal of the Acoustical Society of America*, 92(3):1394–1402, 1992.

[24] J. C. Brown and M. S. Puckette. An efficient algorithm for the calculation of a constant-q transform. *Journal of the Acoustical Society of America*, 92(5), 1992.

[25] J. C. Brown and P. Smaragdis. Independent component analysis for automatic note extraction from musical trills. *Journal of the Acoustical Society of America*, 115, May 2004.

[26] J.-F. Cardoso. Multidimensional independent component analysis. In *Proceedings of IEEE International Conference on Audio, Speech and Signal Processing*, Seattle, USA, 1998.

[27] J.-F. Cardoso. High-order contrasts for independent component analysis. *Neural Computation*, 11(1), 1999.

[28] M. A. Casey. *Auditory Group Theory with Applications to Statistical Basis Methods for Structured Audio*. PhD thesis, Massachusetts Institute of Technology, 1998.

[29] M. A. Casey. MPEG-7 sound-recognition tools. *IEEE Transactions on Circuits and Systems for Video Technology*, 11(6), June 2001.

[30] M. A. Casey and A. Westner. Separation of mixed audio sources by independent subspace analysis. In *Proceedings of International Computer Music Conference*, Berlin, Germany, 2000.

[31] A. T. Cemgil. *Bayesian Music Transcription*. PhD thesis, Radboud University Nijmegen, 2004.

[32] G. Chechik, A. Globerson, M. J. Anderson, E. D. Young, I. Nelken, and N. Tishby. Group redundancy measures reveal redundancy reduction in the auditory pathway. In *Proceedings of Neural Information Processing Systems*, Vancouver, Canada, 2001.

[33] V. Cherkassky and F. Mulier. *Learning From Data: Concepts, Theory and Methods*. John Wiley & Sons, New York, 1998.

[34] M. P. Cooke. *Modeling auditory processing and organisation*. Cambridge University Press, Cambridge, UK, 1993.

[35] L. Daudet and D. Torresani. Sparse adaptive representations for musical signals. In Klapuri and Davy [102].

[36] M. Davies and L. Daudet. Sparse audio representations using the MCLT. *IEEE Signal Processing Letters*, 13(7), 2006.

[37] S. Davis and P. Mermelstein. Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences. *IEEE Transactions on Speech and Audio Processing*, 28(4), 1980.

[38] M. Davy and S. Godsill. Bayesian harmonic models for musical signal analysis. In *Proceedings of Seventh Valencia International meeting Bayesian statistics 7*, Tenerife, Spain, June 2002.

[39] A. de Cheveigné. 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, 1993.

[40] A. de Cheveigné. The auditory system as a separation machine. In A. J. M. Houtsma, A. Kohlrausch, V. F. Prijs, and R. Schoonhoven, editors, *Physiological and Psychophysical Bases of Auditory Function*. Shaker Publishing BV, Maastricht, The Netherlands, 2000.

[41] P. Depalle, G. Garcia, and X. Rodet. Tracking of partials for additive sound synthesis using hidden Markov models. In *Proceedings of IEEE International Conference on Audio, Speech and Signal Processing*, Minneapolis, USA, 1993.

[42] P. Depalle and T. Hélie. Extraction of spectral peak parameters using a short-time Fourier transform modeling and no sidelobe windows. In *Proceedings of IEEE Workshop on Applications of Signal Processing to Audio and Acoustics*, New Paltz, USA, 1997.

[43] P. Depalle and L. Tromp. An improved additive analysis method using parametric modelling of the short-time Fourier transform. In *Proceedings of International Computer Music Conference*, Hong Kong, China, 1996.

[44] DFH Superior. Toontrack Music, 2003. `http://www.toontrack.com/superior.shtml`.

[45] S. Dubnov. Extracting sound objects by independent subspace analysis. In *Proceedings of the 22nd International Audio Engineering Society Conference*, Espoo, Finland, June 2002.

[46] D. E. Dudgeon and R. M. Mersereau. *Multidimensional Digital Signal Processing*. Prentice Hall, Englewood Cliffs, USA, 1984.

[47] D. P. W. Ellis. *Prediction-driven computational auditory scene analysis*. PhD thesis, Massachusetts Institute of Technology, 1996.

[48] D. P. W. Ellis. Evaluating speech separation systems. In P. Divenyi, editor, *Speech Separation by Humans and Machines*, pages 295–304. Kluwer Academic Publishers, 2005.

[49] D. P. W. Ellis and D. F. Rosenthal. Mid-level representations for computational auditory scene analysis. In *Proceedings of International Joint Conference on Artificial Intelligence*, Montreal, Quebec, 1995.

[50] P. A. A. Esquef, V. Välimäki, and M. Karjalainen. Restoration and enhancement of solo guitar recordings based on sound source modeling. *Journal of the Audio Engineering Society*, 50(5), 2002.

[51] M. R. Every. *Separation of musical sources and structure from single-channel polyphonic recordings*. PhD thesis, University of York, 2006.

[52] M. R. Every and J. E. Szymanski. A spectral-filtering approach to music signal separation. In *Proceedings of International Conference on Digital Audio Effects*, Naples, Italy, 2004.

[53] M. R. Every and J. E. Szymanski. Separation of synchronous pitched notes by spectral filtering of harmonics. *IEEE Transactions on Audio, Speech, and Language Processing*, 2006. Accepted for publication.

[54] FastICA package for MATLAB. `http://www.cis.hut.fi/projects/ica/fastica/`, 1st June, 2001.

[55] C. Févotte and S. J. Godsill. A Bayesian approach for blind separation of sparse sources. *IEEE Transactions on Speech and Audio Processing*, 2006. Accepted for publication.

[56] D. FitzGerald. *Automatic Drum Transcription and Source Separation*. PhD thesis, Dublin Institute of Technology, 2004.

[57] D. FitzGerald, E. Coyle, and B. Lawlor. Sub-band independent subspace analysis for drum transcription. In *Proceedings of International Conference on Digital Audio Effects*, Hamburg, Germany, 2002.

[58] D. FitzGerald, E. Coyle, and B. Lawlor. Prior subspace analysis for drum transcription. In *Proceedings of the 114th Audio Engineering Society Convention*, Amsterdam, Netherlands, March 2003.

[59] D. FitzGerald, M. Cranitch, and E. Coyle. Sound source separation using shifted non-negative tensor factorisation. In *Proceedings of IEEE International Conference on Audio, Speech and Signal Processing*, Toulouse, France, 2006.

[60] H. Fletcher, E. D. Blackham, and R. Stratton. Quality of piano tones. *Journal of the Acoustical Society of America*, 34(6), 1962.

[61] N. Fletcher and T. Rossing. *The Physics of Musical Instruments*. Springer, Berlin, Germany, second edition, 1998.

[62] T. Gautama and M. M. Van Hulle. Separation of acoustic signals using self-organizing neural networks. In *Proceedings of IEEE Neural Network for Signal Processing Workshop*, Madison, USA, 1999.

[63] S. Godsill and M. Davy. Bayesian harmonic models for musical pitch estimation and analysis. In *Proceedings of IEEE International Conference on Audio, Speech and Signal Processing*, Orlando, USA, 2002.

[64] G. H. Golub. *Matrix Computations*. Johns Hopkins University press, Baltimore, USA, third edition, 1996.

[65] M. Goodwin. Matching pursuit with damped sinusoids. In *Proceedings of IEEE International Conference on Audio, Speech and Signal Processing*, Munich, Germany, 1997.

[66] M. M. Goodwin. *Adaptive Signal Models: Theory, Algorithms, and Audio Applications*. PhD thesis, University of California, Berkeley, 1997.

[67] M. Goto. A predominant-f0 estimation method for real-world musical audio signals: Map estimation for incorporating prior knowledge about f0s and tone models. In *Proceedings of Workshop on Consistent & Reliable Acoustic Cues for Sound Analysis*, 2001.

[68] M. Goto. Music scene description. In Klapuri and Davy [102], pages 327–359.

[69] M. Goto, H. Hashiguchi, T. Nishimura, and R. Oka. RWC music database: Popular, classical, and jazz music databases. In *Proceedings of International Conference on Music Information Retrieval*, Paris, France, 2002.

[70] R. Gribonval and E. Bacry. Harmonic decomposition of audio signals with matching pursuit. *IEEE Transactions on Signal Processing*, 51(1):101–111, 2003.

[71] D. Griffin and J. Lim. Signal estimation from modified short-time Fourier transform. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 32:236–242, 1984.

[72] S. Handel. Timbre perception and auditory object identification. In Moore [124], pages 425–261.

[73] A. Härmä, M. Karjalainen, L. Savioja, V. Välimäki, U. K. Laine, and J. Huopaniemi. Frequency-warped signal processing for audio applications. *Journal of the Audio Engineering Society*, 48(11), 2000.

[74] F. J. Harris. On the use of windows for harmonic analysis with the discrete Fourier transform. *Proceedings of the IEEE*, 66(1), 1978.

[75] W. M. Hartmann. *Signals, sound, and sensation*. Springer, New York, 1998.

[76] T. Hastie, R. Tibshirani, and J. Friedman. *The Elements of Statistical Learning — Data Mining, Inference, and Prediction*. Springer, 2001.

[77] S. Haykin. *Unsupervised Adaptive Filtering — Volume 1: Blind Source Separation*. John Wiley & Sons, New York, 2000.

[78] M. Helén and T. Virtanen. Perceptually motivated parametric representation for harmonic sounds for data compression purposes. In *Proceedings of International Conference on Digital Audio Effects*, London, UK, 2003.

[79] M. Helén and T. Virtanen. Separation of drums from polyphonic music using non-negtive matrix factorization and support vector machine. In *Proceedings of European Signal Processing Conference*, Turkey, 2005.

[80] R. V. Hogg and A. T. Craig. *Introduction to Mathematical Statistics*. Macmillan Publishing Co., Inc., fourth edition, 1989.

[81] P. Hoyer. Non-negative sparse coding. In *Proceedings of IEEE Workshop on Networks for Signal Processing XII*, Martigny, Switzerland, 2002.

[82] P. O. Hoyer. Non-negative matrix factorization with sparseness constraints. *Journal of Machine Learning Research*, 5:1457–1469, 2004.

[83] G. Hu and D. L. Wang. Monaural speech segregation based on pitch tracking and amplitude modulation. *IEEE Transactions on Neural Networks*, 15(5), 2004.

[84] A. Hyvärinen. Fast and robust fixed-point algorithms for independent component analysis. *IEEE Transactions on Neural Networks*, 10(3):626–634, 1999.

[85] A. Hyvärinen and P. Hoyer. Emergence of phase and shift invariant features by decomposition of natural images into independent feature subspaces. *Neural Computation*, 12(7):1705–1720, 2000.

[86] A. Hyvärinen, J. Karhunen, and E. Oja. *Independent Component Analysis*. John Wiley & Sons, 2001.

[87] International Organization for Standardization. *ISO/IEC 13818-7:1997: Generic Coding of Moving Pictures and Associated Audio Information — Part 7: Advanced Audio Coding*. International Organization for Standardization, Geneva, Switzerland, 1997.

[88] International Organization for Standardization. *ISO/IEC 14496-3:1999: Information technology — Coding of audio-visual objects — Part 3: Audio*. International Organization for Standardization, Geneva, Switzerland, 1999.

[89] The University of Iowa Musical Instrument Samples Database. `http://theremin.music.uiowa.edu`, 1st July, 2000.

[90] G.-J. Jang and T.-W. Lee. A maximum likelihood approach to single channel source separation. *Journal of Machine Learning Research*, 23:1365 – 1392, 2003.

[91] J. Jensen and J. H. L. Hansen. Speech enhancement using a constrained iterative sinusoidal model. *IEEE Transactions on Speech and Audio Processing*, 9(8), 2001.

[92] M. Karjalainen and T. Tolonen. Separation of speech signals using iterative multipitch analysis and prediction. In *Proceedings of 6th European Conf. Speech Communication and Technology*, pages 2187–2190, Budapest, Hungary, 1999.

[93] K. Kashino. Auditory scene analysis in music signals. In Klapuri and Davy [102].

[94] K. Kashino, K. Nakadai, T. Kinoshita, and H. Tanaka. Organisation of hierarchical perceptual sounds: Music scene analysis with autonomous processing modules and a quantitative information integration mechanism. In *Proceedings of International Joint Conference on Artificial Intelligence*, pages 158–164, Montreal, Quebec, 1995.

[95] K. Kashino and H. Tanaka. A sound source separation system with the ability of automatic tone modeling. In *Proceedings of International Computer Music Conference*, pages 248–255, Hong Kong, China, 1993.

[96] M. Kay. *Modern Spectral Estimation*. Prentice Hall, 1988.

[97] S. M. Kay. *Fundamentals of Statistical Signal Processing: Estimation Theory*. Prentice Hall, 1993.

[98] H.-G. Kim, J. J. Burred, and T. Sikora. How efficient is MPEG-7 for general sound recognition? In *Proceedings of the 25th International Audio Engineering Sociery Conference Metadata for Audio*, London, UK, 2004.

[99] A. Klapuri. Sound onset detection by applying psychoacoustic knowledge. In *Proceedings of IEEE International Conference on Audio, Speech and Signal Processing*, Phoenix, USA, 1999.

[100] A. Klapuri. Multipitch estimation and sound separation by the spectral smoothness principle. In *Proceedings of IEEE International Conference on Audio, Speech and Signal Processing*, Salt Lake City, USA, 2001.

[101] A. Klapuri. Auditory-model based methods for multiple fundamental frequency estimation. In Klapuri and Davy [102], pages 229–265.

[102] A. Klapuri and M. Davy, editors. *Signal Processing Methods for Music Transcription*. Springer, 2006.

[103] A. Klapuri, T. Virtanen, and M. Helén. Modeling musical sounds with an interpolating state model. In *Proceedings of European Signal Processing Conference*, Turkey, 2005.

[104] A. Klapuri, T. Virtanen, and J.-M. Holm. Robust multipitch estimation for the analysis and manipulation of polyphonic musical signals. In *Proceedings of International Conference on Digital Audio Effects*, Verona, Italy, 2000.

[105] A. P. Klapuri. Multiple fundamental frequency estimation based on harmonicity and spectral smoothness. *IEEE Transactions on Speech and Audio Processing*, 11(6):804–815, 2003.

[106] A. P. Klapuri. *Signal processing methods for the automatic transcription of music*. PhD thesis, Tampere university of technology, 2004.

[107] T. Kohonen. Emergence of invariant-feature detectors in the adaptive-subspace self-organizing map. *Biological Cybernetics*, 75:281 – 291, 1996.

[108] K. Kreutz-Delgado, J. F. Murray, B. D. Rao, K. Engan, T.-W. Lee, and T. J. Sejnowski. Dictionary learning algorithms for sparse representation. *Neural Computation*, 15:349–396, 2003.

[109] A. Kufner and J. Kadler. *Fourier Series*. Iliffe Books, London, 1971.

[110] J. Laroche, Y. Stylianou, and E. Moulines. HNS: Speech modification based on a harmonic + noise model. In *Proceedings of IEEE International Conference on Audio, Speech and Signal Processing*, Minneapolis, USA, 1993.

[111] C. L. Lawson and R. J. Hanson. *Solving Least Squares Problems*. Prentice Hall, Englewood Cliffs, New Jersey, 1974.

[112] C. K. Lee and D. G. Childers. Cochannel speech separation. *Journal of the Acoustical Society of America*, 83(1), 1988.

[113] D. D. Lee and H. S. Seung. Learning the parts of objects by non-negative matrix factorization. *Nature*, 401:788–791, October 1999.

[114] D. D. Lee and H. S. Seung. Algorithms for non-negative matrix factorization. In *Proceedings of Neural Information Processing Systems*, pages 556–562, Denver, USA, 2001.

[115] P. Lepain. Polyphonic pitch extraction from musical signals. *Journal of New Music Research*, 28(4):296–309, 1999.

[116] S. N. Levine. *Audio Representations for Data Compression and Compressed Domain Processing*. PhD thesis, Stanford University, 1998.

[117] C.-J. Lin. On the convergence of multiplicative update algorithms for non-negative matrix factorization. *IEEE Transactions on Neural Networks*, 2005. submitted for publication.

[118] R. Maher and J. Beauchamp. Fundamental frequency estimation of musical signals using a two-way mismatch procedure. *Journal of the Acoustical Society of America*, 95(4):2254–2263, Apr. 1994.

[119] R. C. Maher. Evaluation of a method for separating digitized duet signals. *Journal of the Acoustical Society of America*, 38(12), 1990.

[120] R. J. McAulay and T. F. Quatieri. Speech analysis/synthesis based on a sinusoidal representation. *IEEE Transactions on Speech and Audio Processing*, 34(4), 1986.

[121] R. Meddis and M. J. Hewitt. Virtual pitch and phase sensitivity of a computer model of the auditory periphery. I: Pitch identification. II: Phase sensitivity. *Journal of the Acoustical Society of America*, 89(6):2866–2894, 1991.

[122] R. Meddis and L. O'Mard. A unitary model of pitch perception. *Journal of the Acoustical Society of America*, 102(3):1811–1820, 1997.

[123] P. Mermelstein. Evaluation of a segmental SNR measure as an indicator of the quality of ADPCM coded speech. *Journal of the Acoustical Society of America*, 66(6), 1979.

[124] B. C. J. Moore, editor. *Hearing—Handbook of Perception and Cognition*. Academic Press, San Diego, California, second edition, 1995.

[125] J. Nieuwenhuijse, R. Heusdens, and E. F. Deprettere. Robust exponential modeling of audio signals. In *Proceedings of IEEE International Conference on Audio, Speech and Signal Processing*, Seattle, USA, 1998.

[126] M. Nishiguchi, K. Iijima, and J. Matsumoto. Harmonic vector excitation coding of speech at 2.0 kbps. In *Proceedings of IEEE Workshop on Speech Coding For Telecommunications Proceeding*, Pocono Manor, USA, 1997.

[127] T. Nomura, M. Iwadare, M. Serizawa, and K. Ozawa. A bit rate and bandwidth scalable CELP coder. In *Proceedings of IEEE International Conference on Audio, Speech and Signal Processing*, Seattle, USA, 1998.

[128] B. A. Olshausen. Sparse codes and spikes. In R. P. N. Rao, B. A. Olshausen, and M. S. Lewicki, editors, *Probabilistic Models of the Brain: Perception and Neural Function*, pages 257–272. MIT Press, 2002.

[129] B. A. Olshausen. Learning sparse, overcomplete representations of time-varying natural images. In *Proceedings of IEEE International Conference on Image Processing*, Barcelona, Spain, 2003.

[130] B. A. Olshausen and D. F. Field. Sparse coding with an overcomplete basis set: A strategy employed by V1? *Vision Research*, 37:3311–3325, 1997.

[131] F. Opolko and J. Wapnick. McGill University Master Samples. Technical report, McGill University, Montreal, Canada, 1987.

[132] A. Ozerov, P. Philippe, R. Gribonval, and F. Bimbot. One microphone singing voice separation using source-adapted models. In *Proceedings of IEEE Workshop on Applications of Signal Processing to Audio and Acoustics*, New Paltz, USA, 2005.

[133] P. Paatero and U. Tapper. Positive matrix factorization: a non-negative factor model with optimal utilization of error estimates of data values. *Environmetrics*, 5:111–126, 1994.

[134] T. Parsons. Separation of speech from interfering speech by means of harmonic selection. *Journal of the Acoustical Society of America*, 60(4), 1976.

[135] M. Parviainen and T. Virtanen. Two-channel separation of speech using direction-of-arrival estimation and sinusoids plus transients modeling. In *Proceedings of IEEE International Symposium on Intelligent Signal Processing and Communication Systems*, Awaji Island, Japan, 2003.

[136] J. Paulus and T. Virtanen. Drum transcription with non-negative spectrogram factorisation. In *Proceedings of European Signal Processing Conference*, Turkey, 2005.

[137] F. Pereira and T. Ebrahimi. *The MPEG-4 Book*. Prentice Hall, Upper Saddle River, 2002.

[138] M. D. Plumbley. Conditions for non-negative independent component analysis. *IEEE Signal Processing Letters*, 9(6), 2002.

[139] M. D. Plumbley and E. Oja. A 'non-negative PCA' algorithm for independent component analysis. *IEEE Transactions on Neural Networks*, 15(1):66–67, 2004.

[140] W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery. *Numerical Recipes in C: The Art of Scientific Computing*. Cambridge University Press, New York, USA, second edition, 1992.

[141] H. Purnhagen and N. Meine. HILN: the MPEG-4 parametric audio coding tools. In *Proceedings of IEEE International Symposium on Circuits and Systems*, Geneva, Switzerland, 2000.

[142] T. F. Quatieri and R. G. Danisewicz. An approach to co-channel talker interference suppression using a sinusoidal model for speech. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 38(1), 1990.

[143] B. Raj and P. Smaragdis. Latent variable decomposition of spectrograms for single channel speaker separation. In *Proceedings of IEEE Workshop on Applications of Signal Processing to Audio and Acoustics*, New Paltz, USA, 2005.

[144] A. M. Reddy and B. Raj. Soft mask estimation for single channel speaker separation. In *Proceedings of ISCA Tutorial and Research Workshop on Statistical and Perceptual Audio Processing*, Jeju, Korea, 2004.

[145] X. Rodet. Musical sound signal analysis/synthesis: Sinusoidal+residual and elementary waveform models. In *Proceedings of IEEE Time-Frequency and Time-Scale Workshop*, Coventry, UK, 1997.

[146] T. D. Rossing. *The Science of Sound*. Addison Wesley, second edition, 1990.

[147] S. Roweis. One microphone source separation. In T. K. Leen, T. G. Dietterich, and V. Tresp, editors, *Proceedings of Neural Information Processing Systems*, pages 793–799, Denver, USA, 2000.

[148] W. Rudin. *Real and Complex Analysis*. McGraw-Hill, third edition, 1987.

[149] M. Ryynänen and A. Klapuri. Polyphonic music transcription using note event modeling. In *Proceedings of IEEE Workshop on Applications of Signal Processing to Audio and Acoustics*, New Paltz, USA, 2005.

[150] D. Sarkar. Methods to speed up error back-propagation learning algorithms. *ACM Computing Surveys*, 27(4), 1995.

[151] L. K. Saul, F. Sha, and D. D. Lee. Statistical signal processing with nonnegativity constraints. In *Proceedings of 8th European Conf. on Speech Communication and Technology*, Geneva, Switzerland, 2003.

[152] E. D. Scheirer. Structured audio and effects processing in the MPEG-4 multimedia standard. *Computer Science*, 7(1), 1999.

[153] E. D. Scheirer. *Music-Listening Systems*. PhD thesis, Massachusetts Institute of Technology, 2000.

[154] M. N. Schmidt and M. Mørup. Nonnegative matrix factor 2-D deconvolution for blind single channel source separation. In *Proceedings of the 6th International Symposium on Independent Component Analysis and Blind Signal Separation*, Charleston, USA, 2006.

[155] M. N. Schmidt and R. K. Olsson. Single-channel speech separation using sparse non-negative matrix factorization. In *Proceedings of the 9th International Conference on Spoken Language Processing*, Pittsburgh, USA, 2006.

[156] D. Schobben, K. Torkkola, and P. Smaragdis. Evaluation of blind signal separation methods. In *Proceedings of International Conference on Independent Component Analysis and Signal Separation*, Aussois, France., 1999.

[157] X. Serra. *A System for Sound Analysis/Transformation/Synthesis Based on a Deterministic Plus Stochastic Decomposition*. PhD thesis, Dept. of Music, Stanford University, 1989.

[158] X. Serra. Musical sound modeling with sinusoids plus noise. In C. Roads, S. Pope, A. Picialli, and G. D. Poli, editors, *Musical Signal Processing*, pages 91–122. Swets & Zeitlinger, 1997.

[159] F. Sha and L. K. Saul. Real-time pitch determination of one or more voices by nonnegative matrix factorization. In *Proceedings of Neural Information Processing Systems*, Vancouver, Canada, 2004.

[160] J. Signés, Y. Fisher, and A. Eleftheriadis. MPEG-4's binary format for scene description. *Signal Processing: Image Communication*, 15(45), 2000.

[161] J. Sillanpää, A. Klapuri, J. Seppänen, and T. Virtanen. Recognition of acoustic noise mixtures by combined bottom-up and top-down processing. In *Proceedings of European Signal Processing Conference*, Tampere, Finland, 2000.

[162] M. Slaney, D. Naar, and R. F. Lyon. Auditory model inversion for sound separation. In *Proceedings of IEEE International Conference on Audio, Speech and Signal Processing*, Adelaide, Australia, 1994.

[163] P. Smaragdis. *Redundancy Reduction for Computational Audition, a Unifying Approach*. PhD thesis, Massachusetts Institute of Technology, 1997.

[164] P. Smaragdis. Discovering auditory objects through non-negativity constraints. In *Proceedings of ISCA Tutorial and Research Workshop on Statistical and Perceptual Audio Processing*, Jeju, Korea, 2004.

[165] P. Smaragdis. Non-negative matrix factor deconvolution; extraction of multiple sound sources from monophonic inputs. In *Proceedings of the 5th International Symposium on Independent Component Analysis and Blind Signal Separation*, Granada, Spain, September 2004.

[166] P. Smaragdis and J. C. Brown. Non-negative matrix factorization for polyphonic music transcription. In *Proceedings of IEEE Workshop on Applications of Signal Processing to Audio and Acoustics*, New Paltz, USA, 2003.

[167] J. Smith and X. Serra. PARSHL: An analysis/synthesis program for non-harmonic sounds based on a sinusoidal representation. In *Proceedings of International Computer Music Conference*, Urbana, USA, 1987.

[168] *Proceedings of Signal Processing with Adaptative Sparse Structured Representations 2005*, Rennes, France, 2005.

[169] A. D. Sterian. *Model-based segmentation of time-frequency images for musical transcription*. PhD thesis, University of Michigan, 1999.

[170] P. Stoica and R. L. Moses. *Introduction to Spectral Analysis*. Prentice Hall, 1997.

[171] P. Stoica and A. Nehorai. Statistical analysis of two nonlinear least-squares estimators of sine-wave parameters in the colored-noise case. *Circuits, Systems, and Signal Processing*, 8(1), 1989.

[172] J. V. Stone, J. Porrill, C. Buchel, and K. Friston. Spatial, temporal, and spatiotemporal independent component analysis of fMRI data. In *Proceedings of the 18th Leeds Statistical Research Workshop on Spatial-Temporal Modelling and its Applications*, 1999.

[173] T. Tolonen. Methods for separation of harmonic sound sources using sinusoidal modeling. In *Proceedings of the 106th Audio Engineering Society Convention*, Munich, Germany, 1999.

[174] T. Tolonen. *Object-Based Sound Source Modeling*. PhD thesis, Helsinki University of Technology, 2000.

[175] T. Tolonen and M. Karjalainen. A computationally efficient multipitch analysis model. *IEEE Transactions on Speech and Audio Processing*, 8(6):708–716, 2000.

[176] K. Torkkola. Blind separation for audio signals — are we there yet? In *Proceedings of International Conference on Independent Component Analysis and Signal Separation*, Aussois, France., 1999.

[177] K. Torkkola. Blind separation of delayed and convolved sources. In S. Haykin, editor, *Unsupervised Adaptive Filtering — Volume 1: Blind Source Separation*, pages 321–375. John Wiley & Sons, 2000.

[178] C. Uhle, C. Dittmar, and T. Sporer. Extraction of drum tracks from polyphonic music using independent subspace analysis. In *Proceedings of the 4th International Symposium on Independent Component Analysis and Blind Signal Separation*, Nara, Japan, 2003.

[179] S. V. Vaseghi. *Advanced Digital Signal Processing and Noise Reduction*. John Wiley & Sons, second edition, 2000.

[180] T. S. Verma. *A Perceptually Based Audio Signal Model with Application to Scalable Audio Compression*. PhD thesis, Stanford University, 1999.

[181] E. Vincent. Musical source separation using time-frequency source priors. *IEEE Transactions on Audio, Speech, and Language Processing*, 14(1), 2006.

[182] E. Vincent, R. Gribonval, and C. Fevotte. Performance measurement in blind audio source separation. *IEEE Transactions on Audio, Speech, and Language Processing*, 2006. Accepted for publication.

[183] E. Vincent and M. D. Plumbley. A prototype system for object coding of musical audio. In *Proceedings of IEEE Workshop on Applications of Signal Processing to Audio and Acoustics*, New Paltz, USA, 2005.

[184] E. Vincent and X. Rodet. Music transcription with ISA and HMM. In *Proceedings of the 5th International Symposium on Independent Component Analysis and Blind Signal Separation*, Granada, Spain, 2004.

[185] T. Virtanen. Separation of harmonic sound sources using sinusoidal modeling. In *Proceedings of IEEE International Conference on Audio, Speech and Signal Processing*, Instanbul, Turkey, 2000.

[186] T. Virtanen. Accurate sinusoidal model analysis and parameter reduction by fusion of components. In *Proceedings of the 110th Audio Engineering Society Convention*, Amsterdam, Netherlands, 2001.

[187] T. Virtanen. Audio signal modeling with sinusoids plus noise. Master's thesis, Tampere University of Technology, 2001.

[188] T. Virtanen. Algorithm for the separation of harmonic sounds with time-frequency smoothness constraint. In *Proceedings of International Conference on Digital Audio Effects*, London, UK, 2003.

[189] T. Virtanen. Sound source separation using sparse coding with temporal continuity objective. In *Proceedings of International Computer Music Conference*, Singapore, 2003.

[190] T. Virtanen. Separation of sound sources by convolutive sparse coding. In *Proceedings of ISCA Tutorial and Research Workshop on Statistical and Perceptual Audio Processing*, Jeju, Korea, 2004.

[191] T. Virtanen. Monaural sound source separation by non-negative matrix factorization with temporal continuity and sparseness criteria. *IEEE Transactions on Audio, Speech, and Language Processing*, 2006. Accepted for publication.

[192] T. Virtanen. Speech recognition using factorial hidden Markov models for separation in the feature space. In *Proceedings of the 9th International Conference on Spoken Language Processing*, Pittsburgh, USA, 2006.

[193] T. Virtanen. Unsupervised learning methods for source separation in monaural music signals. In Klapuri and Davy [102], pages 267–296.

[194] T. Virtanen and A. Klapuri. Separation of harmonic sounds using multipitch analysis and iterative parameter estimation. In *Proceedings of IEEE Workshop on Applications of Signal Processing to Audio and Acoustics*, New Paltz, USA, 2001.

[195] T. Virtanen and A. Klapuri. Separation of harmonic sounds using linear models for the overtone series. In *Proceedings of IEEE International Conference on Audio, Speech and Signal Processing*, Orlando, USA, 2002.

[196] H. Viste. *Binaural Localization and Separation Techniques*. PhD thesis, Ecole Polytechnique Fédérale de Lausanne, 2004.

[197] H. Viste and G. Evangelista. A method for separation of overlapping partials based on similarity of temporal envelopes in multi-channel mixtures. *IEEE Transactions on Audio, Speech, and Language Processing*, 3(14), 2006.

[198] V. Välimäki, J. Huopaniemi, M. Karjalainen, and Z. Janosy. Physical modeling of plucked string instruments with application to real-time sound synthesis. *Journal of the Audio Engineering Society*, 44(5), 1996.

[199] V. Välimäki, M. Ilmoniemi, and M. Huotilainen. Decomposition and modification of musical instrument sounds using a fractional delay allpass filter. In *Proceedings of the 6th Nordic Signal Processing Symposium*, Espoo, Finland, 2004.

[200] P. Walmsley, S. J. Godsill, and P. J. W. Rayner. Multidimensional optimisation of harmonic signals. In *Proceedings of European Signal Processing Conference*, Island of Rhodes, Greece, Sept. 1998.

[201] A. L.-C. Wang. *Instantaneous and Frequency-Warped Signal Processing Techniques for Auditory Source Separation*. PhD thesis, Stanford University, 1994.

[202] M. Weintraub. The GRASP sound separation system. In *Proceedings of IEEE International Conference on Audio, Speech and Signal Processing*, San Diego, California, 1984.

[203] G. K. Yates. Cochlear structure and function. In Moore [124], pages 41–74.

[204] E. Zwicker and H. Fastl. *Psychoacoustics: Facts and Models*. Springer-Verlag, 1990.

[205] Özgür Yilmaz and S. Rickard. Blind separation of speech mixtures via time-frequency masking. *IEEE Transactions on Signal Processing*, 52(7), 2004.