Music Signal Processing Exploiting Spectral Fluctuation of Singing Voice Using Harmonic/Percussive Sound Separation

（調波音打楽器音分離による 歌声のスペクトルゆらぎに基づく 音楽信号処理の研究）

Hideyuki Tachibana

橘 秀幸

Abstract

Singing voice is one of the most impressive components in music signals. Extracting singing voice from mixed music signals has significance because of the potential needs for content-based music search and potential applicability to a technical component of interactive music players. The thesis contains the descriptions of novel music signal processing techniques on singing voice, especially singing voice enhancement, pitch estimation of singing melody, automatic real-time audio-to-audio karaoke generation system (singing voice suppression), discrimination of speech and singing voice, and some subsidiary technical discussions on fundamental techniques.

The thesis specifically focuses on the spectral fluctuation of singing voice, which is one of the principal characteristics of singing voice along with harmonicity and singing formants (characteristic spectral envelope), etc. Although there have been many studies on singing voice extraction exploiting harmonicity and spectral envelope, as well as many studies on singing voice detection and some related discrimination tasks exploiting singing fluctuation, the fluctuation of singing voice has not necessarily been explicitly exploited in the literature of source separation despite its significance. This thesis describes a promising approach to the fluctuation of singing voice for source separation. Since these properties of singing voice are supposed to be “independent” or “orthogonal” each other, it is supposed that joint use of these properties may enrich the toolbox of singing signal processing techniques.

In order to capture the fluctuation of singing voice, the thesis first characterizes a music signal component into three typical classes; harmonic (quasi-stationary, narrowband), fluctuating (intermediate), and percussive (non-stationary, wideband). The thesis first shows a separation technique of harmonic and percussive sound separation (HPSS) ignoring the fluctuating component for simplicity, then considers the extension of the approach of HPSS to handle the intermediate component, namely the fluctuating component, under the same frame work. The idea here is the use of two differently-resolved spectrograms, one of which has rich temporal resolution and poor frequency resolution, while the other has the opposite resolution. That is, the idea is that the behavior of intermediate component is dependent

on the time-scale of spectrogram on which HPSS is executed. Indeed, the spectral shapes of singing voice are quite different on these two spectrograms, because of the fluctuation. On the former spectrogram the shapes of singing voices are similar to those of sustained harmonic instruments, while on the latter spectrogram it is more similar to those of percussions. On the basis of the idea above, this thesis describes a novel singing voice enhancement technique, which is called two-stage HPSS. The experimental evaluations show that SDR improvement, a commonly-used criterion on the singing voice enhancement, indicated around 4 dB, which is a considerably higher level than some existing methods. The result shows the effectiveness of this approach. The idea beneath the technique is the most important contributions of the dissertation.

In addition to singing voice enhancement, two-stage HPSS is applied to following two problems, both of which are of importance in music information retrieval and music applications, respectively; estimation of the fundamental frequency of singing-melody in mixed music signals; singing voice suppression based on two-stage HPSS, and its application to audio-to-audio karaoke system. On the former application, it is verified that two-stage HPSS basically improves the accuracy of pitch estimation of a simple pitch estimation technique. The tandem connection of two-stage HPSS and a simple pitch tracking algorithm, is evaluated in MIREX, an exchange on music information retrieval. The experimental results in MIREX show that the proposal pitch tracking technique is effective especially in low SNR (voice to accompaniment) conditions, comparing to other participants in MIREX. This is possibly because of the preprocessing by two-stage HPSS. This result also proves the effectiveness of two-stage HPSS. On the latter application, it is qualitatively verified that the singing voice suppression based on two-stage HPSS fairly suppresses the singing voice. The quantitative performance in terms of SDR is basically identical to that of singing voice enhancement. In this dissertation the system is actually implemented in C++, and it is verified the system works in real-time, which is advantageous considering the streaming-based client-side applications. Moreover, due to the efficiency of two-stage HPSS, it is verified that the system works even on a net book in real time.

The thesis further discusses the potential applicability of the idea above to characterizing fluctuation on differently-resolved spectrograms, which decompose a signal into many components according to the characteristic time-scale of fluctuation, and discusses a novel audio signal feature, Characteristic Fluctuation Time-scale (CFTS). The feature is applied to speech/singing discrimination, and its effectiveness comparing to MFCC, a standard audio

feature, is described.

The thesis finally descried HPSS in detail, which forms the basis of all the proposal techniques above, and considers some improvement of the technique, in order to make their performance better. Considering long-term relation on spectrogram, as well as the strict reconstructivity constraint, three variants of HPSS are derived. Especially a variant of HPSS which is mentioned as "HPSS 1-B" has an advantage that the global optimality of the solution is guaranteed and its computation is quite efficient. All the reformation above accelerated the computation of HPSS, under some conditions. The experiments on singing voice enhancement using the modified HPSS is also carried out, and it is shown that the performance is not much different in terms of SDR. This result shows that the modified HPSS can accelerate the singing voice enhancement and its applications, without significant loss of separation performance.

In summary, the contribution of this dissertation is as follows. The principal contribution is on "two-stage HPSS" and the underlying idea of exploiting the spectral fluctuation of singing voice in source separation problem using differently-resolved spectrograms, as well as the applications of the technique to audio melody extraction and a karaoke application. A subsidiary contribution includes the extended idea of "two-stage" to "multiple concurrent," which is also promising approach to detection/recognition tasks. Another subsidiary contribution is the fundamental studies on HPSS to improve, namely accelerate, the techniques above.

Acknowledgements

I would like to express my best gratitude to all the people who have supported me during my 10-year student life at the University of Tokyo (UT).

First of all I would like to express my gratitude to the supervisor, Prof. Shigeki Sagayama (UT → NII[⋆1]). I was given many suggestions from him, from the perspective of a seasoned researcher on speech recognition techniques, signal processing and probabilistic inference methods. In addition, the name of the system described in the dissertation, "Euterpe," is given by him. I am also grateful to the co-supervisors, Prof. Nobutaka Ono (UT → NII[⋆2]) and Prof. Hirokazu Kameoka (NTT Lab.,[⋆3] UT) for making detailed comments on the studies. In addition, they were the first contributors to HPSS, a fundamental technique of the thesis, which were formulated by them in around 2007–2008. The idea of HPSS was first proposed and formulated by Prof. Kameoka when he was a PhD student of Sagayama Lab in 2007. The idea was then realized and several different formulations were studied by a master student Ken-ichi Miyamoto and the professors in 2007 and 2008. I am indeed the successor of the series of studies on HPSS after their graduation from the graduate school.

I have greatly benefited from Prof. Kunihiko Mabuchi (UT), who was my supervisor in my undergraduate project on biomedical engineering in 2007–2008, and has been the *pro forma* supervisor since April 2013. I am grateful to him for supporting my kick-off of my career, as well as taking time to process the formalities on student affairs. I am also grateful to Dr. Masataka Goto (AIST[⋆4]), who has been a gracious mentor for novice students and young researchers in the music information processing community, for encouraging me and making many suggestions for my long-term career.

I am very grateful to the developers of the dataset, as well as the evaluators of MIREX (Music Information Retrieval Exchange). These works are indeed quite difficult for an individual student to carry out. Their efforts to enable the experiments of this field really

[⋆1] UT (–03/2013), National Institute of Informatics, Japan (NII) (04/2013–03/2014)

[⋆2] UT (–03/2011), NII (04/2011–)

[⋆3] Nippon Telegraph and Telephone Corporation (NTT), Japan

[⋆4] National Institute of Advanced Industrial Science and Technology, Japan assisted my study. I would like to thank again their invaluable works.

I received the three-year funding for PhD students by JSPS[⋆5], 04/2010–03/2013. This aid, without which I could not have continued to study in the graduate school, really helped my study and the everyday life economically. Special thanks the people who have been committed to the national policy of science and technology to be positive about the non-urgent study of entertainment which may not immediately serve a useful purpose. I sincerely hope the series of the studies, not only mine, but also all the studies in this field, eventually enrich our cultural life.

I am indebted to the co-workers, who are related to this thesis. Parts of Chapter 3 are done with Takuma Ono in 2009, when he was an undergraduate student and I was a teaching assistant for him. Parts of Chapter 4 are similarly done with Yu Mizuno in 2009–2012 as his master thesis. His research theme in his master's course included pitch transposition and tempo conversion, as well as voice conversion and an automatic conductor system. I borrowed some software components from him. Some of his contribution to "Euterpe" system is summarized in Appendix C.

The automatic karaoke system Euterpe, described in Chapter 4, could not be implemented without the support by Takuho Nakano and Dr. Takuya Nishimoto. The Euterpe system was first completed in 2010 in order to demonstrate our two-stage HPSS in the CREST MUSE symposium 2010, which was going to be held just after my coming back to Japan from France where I collaborated with the team of Dr. Emmanuel Vincent at INRIA[⋆6]. In France, I tried to implement the demo software alone in every nights and weekends under the tight schedule, until the incident occurred that the audio device of my computer went wrong suddenly, which disabled me to test my software of audio processing by myself any further. Under this hard situation I asked the lab members in Japan to test and report the behavior of my buggy software. After that, in France I wrote a code, imagining its behavior, and in Japan the lab members tested it. Mr. Nakano, as well as the former assistant professor of the lab., Dr. Nishimoto, were the strongest supporters then.

I am grateful to the members of Lab #1, Dept. Information Physics and Computing (IPC), UT, for making a comfortable environment. I would like to thank Dr. Eita Nakamura (formally with NII), Tomohiko Nakamura, Masato Tsuchiya, and Tatsuma Ishihara, as well

[⋆5] Japan Society for the Promoting of Science (JSPS) KAKENHI Grant Number 22-6961

[⋆6] *Institut National de Recherche en Informatique et en Automatique*, France

as the alumni: Dr. Jonathan Le Roux (now with MERL[*7]), Dr. Stanisław A. Raczyński (INRIA), Dr. Satoru Fukayama (AIST), Dr. Wu Jun (The Chinese Academy of Sciences), Dr. Miquel Espi (NTT Lab.), Kim Tae Hun (Technische Universität Berlin), Emiru Tsunoo (SONY), Yu Kitano (Hitachi Lab., Japan), Tsubasa Tanaka (Tokyo University of the Arts), Masahiro Nakano (NTT Lab.), Yushi Ueda, Iwao Tanuma (Hitachi Lab.), Akinori Takeuchi, Kazuki Ochiai (SONY), Kosuke Suzuki, and lab secretaries Naoko Tanji, Reiko Kaneko and Kumi Masui. I would also like to thank the member of METISS group of INRIA; Dr. Emmanuel Vincent and Dr. Ngoc Q. K. Duong.

I would also like to express my gratitude to the schoolmates of the Department of Mathematical Engineering and Information Physics (MEIP), UT, for having discussions as well as being friendly with me, especially Sho Kamuro, Kazuhito Tanaka, Keisuke Hasegawa, Dr. Yutaka Hori (now with Caltech), Dr. Yasuaki Monnai, Tomohiro Sueishi, Misa Sato, Yuko Zo, Kota Ishihara, and Kentaro Minami. I am also grateful to the peers of the University of Tokyo Orchestra, especially Jun Ueki (Kyushu University, Mathematics), Shu Sawai (UT Mechanical Engineering), Yasunori Kogure (UT Medical School), Yu Takaseki, Haruhiko Ikeda, and Takayuki Hayashi (UT Astronomy), for cheering me up when being depressed, and sometimes having discussions on music with me.

I am also grateful to the friends on the internet I have kept in touch with, including some students studying pure and applied mathematics, and some software engineers. I have invaluable opportunities to share much information on several topics with them. Although most of the knowledge I acquired in the groups were not actually used in this dissertation, the informal online meetings has been an incentive for me to keep learning across a range of mathematical and computer sciences, from theory to practice.

I would also like to express my gratitude to the contributors to the free softwares I used in my study: GNU C++ compiler, GNU R, ggplot2 package, tuneR package, gnuplot, Tcl/Tk, Inkscape, Dia, LaTeX and its style files, GNU Octave, Audacity, Git, Mercurial, Linux, and the others that are explicitly cited in the main text.

Finally I would like to thank my family.

[*7] Mitsubishi Electric Research Lab., USA

Contents

List of Figures

List of Tables

List of Algorithms and Pseudo-codes

List of Acronyms

AME Audio Melody Extraction. 40, 50, 51, 107

ASR automatic speech recognition. 2, 20

CFTS Characteristic Fluctuation Time-scale. 71, 73–76, 106

CMHPSS Concurrent Multiple HPSS. 70, 76, 105

CQT Constant Q Transform. 17, 43, 44, 48, 52, 111, 112

DFT discrete Fourier transform. 11, 12, 22, 98, 110, 121, 122, 125

FFT fast Fourier transform. 110, 125–127

GNSDR Generalized NSDR. 32, 34, 35, 37

HPSS Harmonic/Percussive Sound Separation. 8, 9, 15, 25–28, 30, 31, 34, 37, 58, 60, 62, 63, 65, 66, 68, 70, 71, 76–79, 81, 92, 93, 96, 98, 101, 103–106, 115, 124, 129, 130

MFCC Mel-frequency Cepstral Coefficients. 17, 74–76, 106, 118

MIR Music Information Retrieval. 2–4, 6–9, 14, 38, 39, 52, 104, 105

MIREX Music Information Retrieval Exchange. 40, 41, 46, 50, 51, 105

NMF non-negative matrix factorization. 13, 14, 17, 26, 86, 96, 98, 101, 118

NSDR Normalized SDR. 32, 34, 35, 58, 101

RTISI-LA real-time iterative spectrogram inversion with look-ahead. 121, 124, 125

SAR signal to artifact ratio. 97, 98

SDR signal to distortion ratio. 15, 32, 34, 35, 58, 64, 93, 97, 98

SER signal to error ratio. 124

SIR signal to interference ratio. 97, 98

SNR signal to noise ratio. 34, 35, 52

STFT short-term Fourier transform. 11, 12, 15–17, 28, 30, 31, 58, 59, 68, 71, 79, 80, 90, 101, 111, 128, 129

two-stage HPSS . 7, 8, 32, 34, 35, 38, 39, 42, 43, 46, 48, 51, 52, 54, 58, 60, 103–107

Chapter 1

Introduction

1.1 Background: Music and Information Technology

1.1.1 Toward Computers That Understand Music

Humans, although it depends on their background, have ability to extract much information from songs, just by listening to it. Let us listen to a song, which may not be famous. Search the keyword *"Oedo Nihonbashi"* on the internet, and the readers may find some recordings. Listening to it, many people may realize that the song sounds ethnic, and may also think that it may possibly be a traditional Asian song.

People may also easily guess which kinds of instruments are used, as well as the way they are used, such as "something similar to a guitar is played as an accompaniment." If they are Japanese people, they may be convinced that the song is probably a traditional Japanese song, even though they do not actually know it at all. Japanese speakers may also understand the lyrics of the song partially. People who are musically-trained may dictate the melody easily as *si-do-si — do-mi-do-si — sol-si-mi-do — si. . .* , and some of them also can dictate the accompaniment. People who have musicological background may find an out-of-tune pitch in this song, and may think it is something similar to *musica ficta* in European medieval music. Some opera fans may realize that the song is quite similar to a short phrase in the wedding scene of Puccini's "*Madama Butterfly*," and understand that the composer quoted the folk song in his work, similarly to the case of a popular Chinese song "*Mo Li Hua*" in "*Turandot*".

Let us next consider a case of computers. When we ask a computer to make a comment on a song, what does it answer? A stereotypic robot in fictions may answer as much information as possible, adding a comment that it cannot understand emotions, for example. Nevertheless, in reality, what is difficult is not only such hard problems, but also many information extraction problems from music recordings. To be precise, humans do not have enough knowledge to formulate such techniques that enable computers to do these tasks, namely guessing that a given song is something exotic, dictating the melody of a given song, and guessing that the given song is identical to a subsidiary motif in a famous opera, etc.

These tasks above are what the computers are supposed to do in our blueprint of the computer understanding of music recordings. The situation is very similar to the cases of the automatic speech recognition (ASR), computer vision (CV), natural language processing (NLP) and other many problems in artificial intelligence [129], in which researchers have devoted efforts to make computers to do what humans easily do.

Music Information Retrieval and Content-based Music Search

In addition to our scientific interest for intelligent computers that understand music, there are many potential applications in real-world if these techniques are realized even if partially. For example, if a computer can answer to questions such as “Who is singing the given song?” “Which kind of instruments are used in the song?” “Is the genre of this song rock or jazz or classical?” and “In which language the given song is sung?” etc., it may be used as a technical component for music search engines, that assist listeners to find their favorite songs. Such content-based search techniques are becoming important practically, because of the recent growth of the music community on the web, on which many songs created by Sunday musicians are uploaded. The number of songs is increasing week by week, and it is becoming almost impossible for humans to find their favorite songs in the huge database.

On the basis of the background, many techniques to extract information from music signals have been proposed, in an attempt to apply them to content-based music search in mind, though not limited. The series of studies are called Music Information Retrieval (MIR). For instance, the past decade saw the intense studies on chord estimation [149], melody transcription [14], [25], [27], [43], [47], [65], [66], [81], [82], [124], [130], [134], [305], [316], transcription of polyphonic music [1], [83], genre estimation [93], [148], singer identification [42], lyrics transcription [98], [143], etc.

Some of these techniques are studied in the literature of the signal processing, similarly to the cases of ASR systems, partly because of the similarity of the problems, i.e., both ASR and MIR are the problems on information extraction from the real-world sounds, and signal processing is a suitable technique to handle these kind of problems.

1.1.2 Music Information Processing for Entertainment Applications

Another motivation for music information processing studies comes from the potential applications to a "purpose" of the music itself, namely entertainment. Since music is one of the principal leisure in human activities [163], [164], it is natural to consider applications of the recent progress of information technologies to music, in quest of richer cultural life.

On the basis of recent growth of personal computers, small devices e.g. smart phones, and network community, many concepts and applications on music have been proposed. One of the important concept is the "active music listening," proposed by Goto [50]. This concept aims at the application of the recent information technologies to music listening systems, in which users actively involves in music. For example, using these systems, the listeners can freely change the volume of each instrument in music, and the listeners can freely replace the singer of a song by another singer, etc. Based on the concept, Goto and his collaborators have proposed many active music listening systems e.g., Musicream [48], Drumix [167], Songle [49] and PodCastle [108]. Another remarkable success that exploited recent progress in information technology is Vocaloid [76], a vocal synthesis engine. The Vocaloid songs created by amateur musicians became very popular in late '00s, and now it is considered to be one of the important part of the popular culture of Japan [20], [55]. These applications contrasts with the conventional ways of appreciating music, in which what ordinary listeners can do was just listening to the works created by professional musicians, except for the freedom of the choice of songs.

Behind the brilliant successes above, there lie many sober fundamental technologies which support the applications. The central technology of them is also the signal processing. Indeed, automatic conversion of music signal, discussed in the active music listening, would be a direct descendant of the graphical equalizer, i.e., a bank of band pass filters (BPF).

1.1.3 Technical Interest on Music Signal Processing and Source Separation

As reviewed in the previous two subsections, beneath the applications of music information technology lie the fundamental techniques of signal processing. Among the subfield of signal processing techniques, source separation is one of the important subfield in our purpose. Let us summarize two reasons for the importance, which are related to the discussions above.

- Firstly, it is believed in MIR community that the source separation is an appropriate preprocessing for the MIR tasks above [106]. For example, extracting singing voice from mixed music signals may naturally enhance the subsequent processings such as automatic lyrics transcription, singer identification, melody estimation, etc.
- Another reason is that the separated signals can be directly used in the entertainment applications discussed above, in which users can adjust the volume of vocal sounds and instrumental sounds independently, which has been difficult in an ordinary music players.

Along with the possibilities of the applications, technical difficulties also make music source separation interesting. Let us consider a specific task of singing voice extraction from mixed music signals, which shall be seen in Chapter 2 again. One of the difficulties in singing voice extraction from music signal comes from the similarity between singing voice and accompaniments, e.g., a piano, a guitar. For instance, both the spectra of singing voice and harmonic instruments, such as a piano and a guitar, have harmonic structure. Accordingly, it is difficult for a simple harmonics-extraction technique to detect only the singing voice in polyphonic music signals. Another difficulty is that accompanying instruments do not satisfy some properties of “noise” that have been supposed in conventional signal processing problems, e.g., whiteness and stationarity. Naturally, we cannot expect that classical noise suppression techniques e.g. [8], [31], work effectively in singing voice enhancement, because music signals are not white noise nor stationary.

Development of Feasible Technique under Limited Computation Resource

Despite the scientific and technical importance of music studies described above, however, we may not say that the practical music applications are very serious and we must solve some problems of music as precisely as possible at all costs, using a high performance computer if needed. Instead, the algorithms on practical music applications are required to be efficient. This requirement is not necessarily essential in music signal processing, but desirable practically.

A fair assumption on the computer we may use for practical music applications is that we may only use small personal devices such as smart phones, tablet computers and ordinary personal computers, or ordinary web servers at a maximum. Indeed, some real-world music applications run these computers. For example, Vocaloid [76] runs on personal computer, and an automatic composition system Orpheus [44] runs on an ordinary web server.

Assuming the limited computer resources, it is less favorable that the techniques on music signal processing are based on quite costly operations, such as the use of large-scale matrix multiplications, complicated probabilistic inference which may require Monte-Carlo methods, etc. Instead, it is more preferable that the techniques are mostly comprised of less-costly simple operations, such as element-wise addition of matrices.

1.2 Singing Voice Extraction/Suppression and the Applications

1.2.1 Motivation: Significance of Singing Voice in MIR and Music Applications

In this dissertation, we specifically focus on the problems on singing voice in music, because of the significance of vocal components in music signals. Indeed, In many genres of music, especially in the popular musics, the lead vocal is the most impressive and essential part for most listeners; for example, when people listen to a song, many of them are interested in who sings the song, what is sung in the lyrics of the song, in which language the song is sung, etc., rather than who plays the background accompaniment, who composed the song, etc.

In MIR systems, because of the attentions to the singing voice from the listeners, the techniques related to singing voice may have more significance than those of instruments. Consequently, the queries from users on singing voice may account for the substantial fraction of all the queries in the music search engines. Because of the significance of singing voice in MIR applications, many MIR techniques which principally focused on singing voice have been studied, such as automatic lyrics recognition [98], [99], [143], identification of the language of a song [144], automatic singer identification [42], [79], etc.

In addition to MIR applications, singing voice is also significant in music applications. That is, because of the attentions to the singing voice from many users, vocal-related music applications may be attractive, similarly to existing music applications such as karaoke [165], which is now a very popular way of enjoying music [163], [164], and others that are reviewed in [95].

1.2.2 Singing Voice Enhancement

Thus, considering the significance of the applications that are related to singing voice, it is important to develop techniques to enhance singing voice in music signals. In this dissertation, we first consider the problem of singing voice enhancement in mixed music signals.

In order to extract singing voice from mixed music signals, we naturally need to consider some characteristics of singing voice, in order to distinguish them from accompanying instruments. Sundberg's [142] thorough survey on the studies on singing voice includes some discussions on the characteristics of singing voice. For example, it is discussed in the survey that the singing voice has integer overtones, characteristic formant of the spectrum, etc. Indeed, many singing voice enhancement techniques have been proposed that are based on these features above, namely harmonicity and spectral characteristics (spectral envelope). These existing studies shall be reviewed in Chapter 2.

1.2.3 Fluctuation-based Singing Voice Enhancement

Another important characteristics of singing voice is the vibrato, discussed in [60], [105], [142, Ch.8], etc. There have also been many studies that exploited the fluctuation of singing voice, principally for detection problems, but the use of the fluctuation of singing voice for source separation has not been sufficiently studied up to now despite its significance, except some studies on modulation filtering.

Thus the establishment of the fluctuation-based source separation has importance, because it is not sufficiently studied despite it is a standout nature of singing voice, and because the fluctuation are likely to be an "orthogonal" feature to other well-studied features of singing voice, namely harmonicity and characteristic spectral envelope, which naturally indicates that the joint use of all of them may enhance whole the performance of singing voice extraction.

Because of the reason above, the study to formulate a source separation technique that is principally based on the fluctuation of singing voice is of significance, as a fundamental study for singing voice extraction, and moreover, for the objectives described in the previous section, i.e., MIR and music applications. In this dissertation, we specifically focus on the fluctuation, of singing voice, and discuss a novel approach to enhance singing voice from mixed music signals exploiting the fluctuation. The novel approach is the principal novelty of the dissertation. The detail shall be described in Chapter 2.

1.2.4 Objectives and Contributions of the Thesis

The aim of this dissertation is to develop a novel technique to extract singing voice exploiting its fluctuation, which has not been sufficiently exploited to our best knowledge, and to develop the application of the technique. The contributions of this thesis are as follows.

1. **Develop singing voice enhancement techniques that exploit the spectral fluctuation**

   - We show a novel idea to extract singing voice from music signals, exploiting its spectral fluctuation.
   - We show a singing voice enhancement technique "**two-stage HPSS**" based on the concept.

2. **Solve some application problems of music, exploiting the fluctuation-based techniques above**

   We then apply the method above to following applications.

   (a) **Application to MIR**

   We apply two-stage HPSS to estimate the predominant singing melody from music signals, which is one of the basic MIR tasks.

   (b) **Application to Entertainment Software**

   We apply the two-stage HPSS to an automatic audio-to-audio karaoke generating system, which works in real-time, due to the efficiency of two-stage HPSS.

3. **Extension of the idea of two-stage HPSS to a novel signal decomposition and signal characterization**

   - On the basis of the idea of two-stage HPSS, we propose a novel signal processing technique which decomposes a signal into many components based on the characteristic time-scale of fluctuation, and construct a novel signal feature.
   - We show its application to the discrimination of singing voice and speech.

4. **Describe HPSS in detail, considering some possibilities of reformulation.**

   - We describe the signal processing technique Harmonic/Percussive Sound Separation (HPSS) which forms the basis of this thesis in detail
   - We reformulate HPSS for better performance of the techniques above.

Especially 1 and 2 above form the core contribution of this dissertation, while 3 and 4 are rather subsidiary.

1.3 Thesis Outline

This thesis is organized as shown in Figure 1.1, and each chapter describes the following.

Chapter 2 first describes a characteristics of singing voice. This chapter also describes some fundamental techniques of music signal processing, especially HPSS, which forms the basis of all the techniques in this thesis. This chapter then describes a singing voice enhancement technique in mixed music signals, which is based on the twice application of HPSS on differently-resolved two spectrograms. The idea beneath the technique is one of most important contributions of the dissertation.

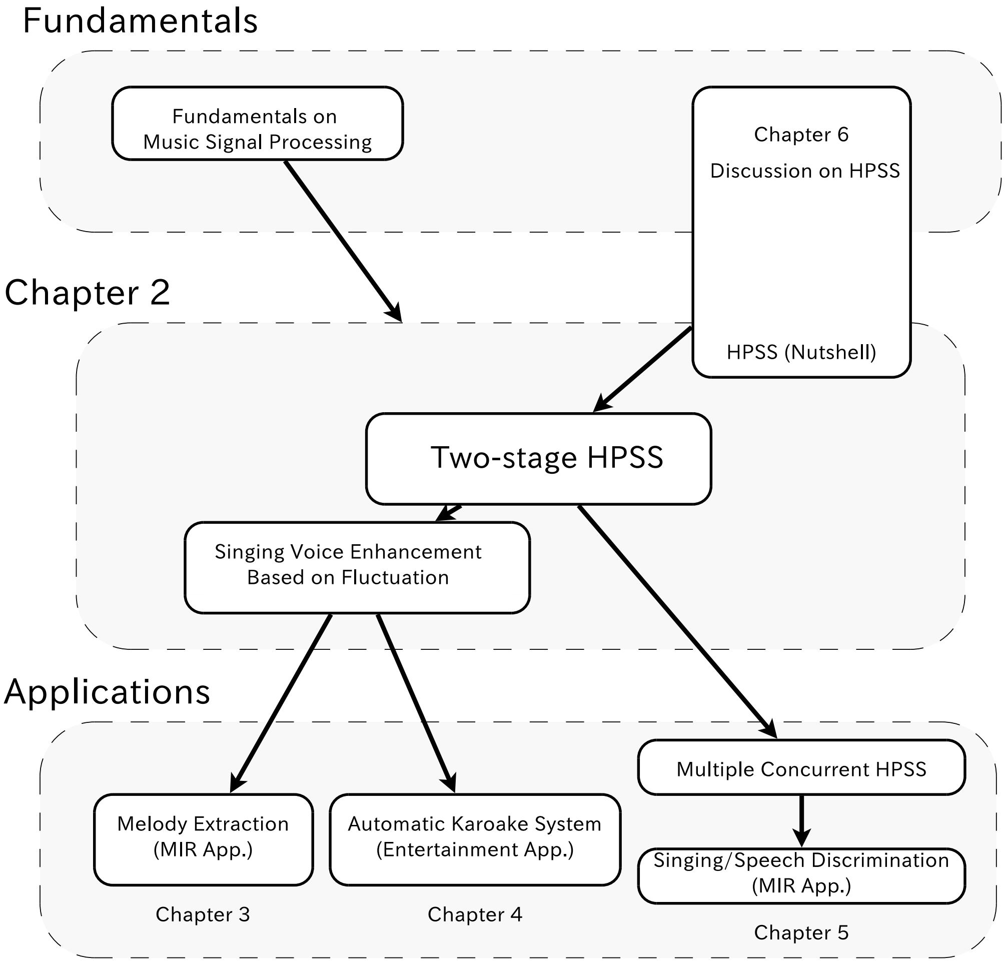

Fundamentals
Fundamentals on Music Signal Processing
Chapter 6
Discussion on HPSS
HPSS (Nutshell)
Chapter 2
Two-stage HPSS
Singing Voice Enhancement Based on Fluctuation
Applications
Multiple Concurrent HPSS
Melody Extraction (MIR App.)
Automatic Karoake System (Entertainment App.)
Singing/Speech Discrimination (MIR App.)
Chapter 3
Chapter 4
Chapter 5

Figure 1.1:: The structure of the dissertation.

Chapter 3 shows an application of the result of Chapter 2 for pitch estimation problems of predominant melodies in mixed music signals, which is one of the basic problems in MIR. Due to the effectiveness of the method discussed in Chapter 2, the pitch estimation technique was also effective comparing to existing methods.

Chapter 4 shows an another application of the technique discussed in Chapter 2 to a real-time audio-to-audio karaoke system.

Chapter 5 shows an extension of the concept of Chapter 2 and show examples of the signal features based on the concept, which may characterize the fluctuation of signals. This chapter also shows an application of the method to singing/speech discrimination.

Chapter 6 describes HPSS in detail, which is the fundamental technique of all the methods above. This chapter also describes reformulations of HPSS.

Finally **Chapter 7** concludes the dissertation.

1.4 Notation

$\mathbb{N}, \mathbb{Z}, \mathbb{R}, \mathbb{C}$ and Floating Point Number

$\mathbb{N}, \mathbb{Z}, \mathbb{R}$ and $\mathbb{C}$ denote natural number, integer, real number, and complex number, respectively, as usual. We define $0 \in \mathbb{N}$ for convenience. In this thesis, all the runtime computations are carried out numerically, and are not dependent on the computer algebra systems. In computing, we always approximated $\mathbb{R}$ by the floating point number **binary32** defined in IEEE 754 (`float` in C/C++, Java, etc., and `real` in Fortran), or **binary64** (`double` in C/C++, Java, etc., `double precision` in Fortran, `real` in R, and `float` in Python). It is the same for $\mathbb{C}$.

Inner Product and Norm

$\langle x(t), y(t)\rangle := \sum_t x(t)y(t)$ denotes Euclidean inner product, and $\|x\| := \sqrt{\langle x, x\rangle}$ denotes the norm.

Division by zero, etc.

In computation, some exceptional cases e.g., "$\ln 0$," "1/0," etc.,[⋆1]are replaced by "$\ln \varepsilon$," "$1/\varepsilon$," etc., where ε denotes a very small non-zero number, e.g., 10^{-100}, which is sufficiently small but sufficiently larger than the smallest positive number of binary64, $\approx 10^{-308}$. The actual programs were written as follows,

$$\frac{x}{y+\varepsilon} \text{ instead of } \frac{x}{y} \tag{1.1}$$

This modification practically does not cause fatal problems.

[⋆1] Note, in IEEE754, it is not defined that the computer must throw exceptions. Hence, even zero division occurred, the computers try to continue calculation forcibly without throwing errors, and they print `NaN` or `Inf` in the end. In order to avoid this inconvenience, C/C++ programmers may use a GNU extension function `feenableexcept()` defined in `fenv.h`, etc., to enable the floating point exception signal when these exceptions occurred. The author recommends to use this optional functions to make debugging procedure easier, because many of the methods in this dissertation contain the operations which can result in `NaN` or `Inf`, unless carefully programmed.

Variables with Superscripts

In this thesis, we sometimes use γ-powered variable "x^γ" instead of "x." For example, $\partial f(x^\gamma)/\partial x^\gamma$, means $\partial f(y)/\partial y$, where $y = x^\gamma$. That is,

$$\frac{\partial}{\partial(x_1^\gamma)} x_1^{m\gamma} x_2 = \frac{\partial}{\partial y} y^m x_2 = m y^{m-1} x_2 = m x_1^{(m-1)\gamma} x_2. \tag{1.2}$$

Similarly, we often use an expression such as $x^\gamma \leftarrow f(x^\gamma)$. This expression is in principle equivalent to $x \leftarrow \sqrt[\gamma]{f(x^\gamma)}$. However, in computation, it is better to evaluate the expression by using a temporary variable y as $y \leftarrow f(y)$, and delay the evaluation of the expression $x \leftarrow \sqrt[\gamma]{y}$ until the very x becomes needed.

Abbreviation of Arguments

We often omit some arguments of functions if evident. For example, $f(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z})$ is sometimes denoted as f for simplicity. When we pay attention to a specific element x_i of the argument $\boldsymbol{x} = (x_1, x_2, \ldots, x_n)$, we denote a function $f(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z})$ as $f(x_i|\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z})$, or simply $f(x_i)$, which mean that we regard only x_i to be a variable, and the other elements of the tuple $\boldsymbol{x}$, i.e., $(x_1, \ldots, x_{i-1}, x_{i+1}, \ldots, x_n)$, as well as $\boldsymbol{y}, \boldsymbol{z}$, are fixed.

STFT

Given a discretized real-valued signal $x(t)$, where t denotes discrete time $t \in \mathbb{Z}$, then short-term Fourier transform (STFT) is denotes as[⋆2]

$$\mathsf{STFT}(L)\,[x(t)] = \tilde{\boldsymbol{X}} = (\tilde{X}_{n,k})_{(n,k)\in\Omega} \tag{1.3}$$

where $L \in \mathbb{N}$ denotes the frame length (size of analyzing window) of STFT. For convenience, we assume L is even as mentioned above. Each element $\tilde{X}_{n,k} \in \mathbb{C}$ is defined by discrete Fourier transform (DFT) (see Appendix A) as follows,

$$\mathsf{STFT}(L)\,[x(t)]_{n,k} := \tilde{X}_{n,k} := \sum_{t=0}^{L-1} x(t + ns - L/2) g_1(t) \exp\{-2\pi\sqrt{-1}tk/L\} \tag{1.4}$$

[⋆2] Other notations may seem more natural, such as $\mathsf{STFT}_L[x(t)]_{n,k}$, $\mathsf{STFT}_{x(t)}^{g_1(t)}(n,k)$ (e.g., [83, Ch. 2]), $T_{n,k}^{\mathrm{win}}(x)$ (e.g., [19, Ch. 1]). However, we noted STFT in this way, because we often pay attention to L in this dissertation, and therefore, this notation is more eye-friendly than others in many situations. To the contrary, we omit $g_1(t)$ from the notation because it is less emphasized on in this dissertation. We may interpret the notation simply as follows; STFT is a higher order function, which takes L as a variable, and returns a function $\mathsf{STFT}(L)[\cdot]$.

where s is the size of frame shift, which is coordinated with L, such as $s = L/2$ in this thesis, and $g_1(t)$ is a window function.

The subscripts $n, k \in \mathbb{Z}$ where $0 \leq k < K := L/2 + 1$ denote the indices of time and frequency, respectively. A pair of n and k, i.e., (n, k) is referred to as "a time-frequency bin," or simply "a bin," which is also referred to as "T-F unit" in some other papers. Note that the values of the spectrogram $\tilde{X}_{n,k}$ outside of $0 \leq n < N$ is $\tilde{X}_{n,k} = 0$ by definition, where $N := \lceil 2f_\mathrm{s}T/L \rceil + 1$. Thus we can write the domain Ω as

$$\Omega = \{(n, k)|n = 0, 1, \ldots, N-1, k = 0, 1, \ldots, K-1\},$$

and regard a complex spectrogram $\tilde{\boldsymbol{X}}$ as an element of $\mathbb{C}^{N \times K}$. The value of the time-frequency bins outside of this domain is defined to be zero for convenience, i.e., $\hat{X}_{n,k} = 0$for any $(n, k) \notin [0, N-1] \times [0, K-1]$.

On STFT and music signal processing, see [83, Sec. 2.1]. For source codes, see Appendix D.1.

Inverse STFT

The inverse STFT, i.e., $\mathsf{STFT}^{-1}(L)[\tilde{\boldsymbol{X}}]$, is defined on the basis of the inverse DFT and the overlap-add (OLA) using a reconstruction window function $g_2(\tau)$ as follows. We first restore the ignored k, i.e., $K \leq k \leq L-1$, by using the rule $\tilde{X}_{n,L-k} = \tilde{X}^*_{n,k}$. Then, we apply the inverse DFT to $(\tilde{X}_{n,k})_{0 \leq n \leq N-1, 0 \leq k \leq L-1}$ to obtain N segments $(x_n(\tau))_{0 \leq n \leq N-1} = (x_1(\tau), x_2(\tau), \ldots, x_n(\tau))$ of length L, where τ is the discrete time $0 \leq \tau \leq L-1$. For convenience, let us define $x_n(\tau) = 0$ when $\tau < 0, L \leq \tau$. We finally apply OLA to synthesize the waveform, using a reconstruction window function $g_2(\tau)$ as follows,

$$x(\tau) = \sum_{n=0}^{N-1} x_n(\tau - (n-1)L/2) g_2(\tau - nL/2), \tag{1.5}$$

and trim away $\tau < 0, f_\mathrm{s}T \leq \tau$. See also Appendix D.1.

Note, it is sometimes believed that it is impossible to reconstruct a signal from the power spectrogram, because the information of phase is lost in power spectrogram. However, in reality, we do not need phase information to synthesize the original signal from the power spectrogram under some conditions. See [18, § 7.9].

For other discussions on time-frequency distribution, see Appendix A.

Arithmetic Operations of Spectrograms

Unlike other methods such as the techniques based on non-negative matrix factorization (NMF), we do not particularly regard a spectrogram $\hat{\boldsymbol{X}}$ as a matrix in this dissertation. Instead, we regard it just a tuple of $N \times K$ complex/real numbers, and define arithmetic operations as element-wise operations. For example,

$$\boldsymbol{X}+\boldsymbol{Y} := (X_{n,k}+Y_{n,k}), \tag{1.6}$$

$$\boldsymbol{X}\boldsymbol{Y} := (X_{n,k}Y_{n,k}), \tag{1.7}$$

$$|\boldsymbol{X}^{\gamma}|/2 := (|X_{n,k}{}^{\gamma}|/2), \tag{1.8}$$

$$\sqrt{\boldsymbol{X}} := (\sqrt{X_{n,k}}), \tag{1.9}$$

$$\boldsymbol{X} \ \geq \ 0 \overset{\text{def}}{\Leftrightarrow} \forall(n,k), X_{n,k} \ \geq \ 0 \tag{1.10}$$

$$e^{\sqrt{-1}\angle\tilde{\boldsymbol{X}}} := (e^{\sqrt{-1}\angle\tilde{X}_{n,k}}) = (\tilde{X}_{n,k}/|\tilde{X}_{n,k}|). \tag{1.11}$$

Chapter 2

Two-stage HPSS: Singing Voice Enhancement Exploiting Fluctuation

2.1 Introduction

In many genres of music, especially in the popular musics, the lead vocal is the most impressive and essential part for most listeners, and moreover, it often has much information that is important in MIR applications. In fact, many MIR studies, such as automatic lyrics recognition [99], [98], [143], identification of the language of a song [144], automatic singer identification [42], etc., have used the information on singing voices. In addition to the importance as a preprocessing for MIR applications, furthermore, it is also significant in itself in the way that the technique can be applied as a kind of interactive music player, i.e., a vocal/nonvocal equalizer, an automatic karaoke generator [132] and etc. Along with the possibilities of the applications, as mentioned in Chapter 1, its technical difficulties also make singing voice enhancement an interesting problem.

In this chapter, we focus on the fluctuation of singing voice, such as vibrato [60], [105], [142, Ch. 8]. In order to capture the fluctuation, we exploit two spectrogram representations with different time-frequency resolutions, which are unlike the existing methods. Our motivation

Notes on Chapter 2: This chapter is a revision of an authors' paper [302], and is the extended version of the authors' previous conference papers [305], [316]. Note, after the conference, Hsu et al. [65] have developed more effective singing voice enhancement technique on the basis of the concept of the authors' previous conference papers [305], [316] as well as their pitch estimation techniques. Another note is that a similar idea was also mentioned by FitzGerald [36] around the same time as our previous works [305], [316]. The idea of using differently-resolved two spectrograms is studied also in NMF framework by Zhu et al. [169] referring the authors' previous conference paper [305].

for using two different spectrograms comes from our observation that singing voice has an "intermediate" property between other harmonic instruments and percussive instruments. That is, a singing voice appears similarly to harmonic instruments on an ordinary spectrogram that has 10–30 [ms] temporal resolution, while it should appear rather similar to percussions if the analyzing frame of STFT is much longer than the temporal scale of the fluctuation of singing voice. On the basis of the idea above, we roughly define three types of musical components – fluctuating, sustained, and transient – and show that those three types of components can be separated by applying a simple source separation technique twice, which is called HPSS, on differently-resolved spectrograms (Section 2.5.) According to the experiments we conducted in order to evaluate the performance of the method compared with those of existing methods, it is verified that the method extracts singing voice effectively, indicating around 4 dB signal to distortion ratio (SDR) [51], [115] improvement, which is a considerably higher level than the other methods.

2.1.1 Related Work: State-of-the-art Singing Voice Extraction Techniques

Because of the many potential applications as well as the technical interests described above, many methods on singing voice enhancement in music signals, and other related techniques including singing melody transcription (see section 3.1.1), have been actively studied recently.

In most of the existing methods, an input music signal is first transformed from time domain to time-frequency domain, then singing voice is characterized there. Many of the techniques especially consider the power spectrogram, the squared amplitude of the STFT, as a time-frequency distribution of the signal. On time-frequency distribution, components other than singing voice, such as accompanying instruments, are suppressed with time-frequency masking (adaptive Wiener filtering), and finally, the estimated spectrogram of singing voice is transformed back to time domain again. That is, a typical technique is based on the processing such as following.

1. Given a signal $y(t)$, and convert it into complex spectrogram $\hat{\boldsymbol{Y}} \in \mathrm{C}^{N\times K}$ by STFT.

2. Take the power of the complex spectrogram $\boldsymbol{Y}^2 = |\hat{\boldsymbol{Y}}|^2$, while the phase spectrogram $\boldsymbol{\phi} = \angle\hat{\boldsymbol{Y}}$ is also stored.
3. Apply some processings to $\boldsymbol{Y}^2$, and obtain another spectrogram $\boldsymbol{V}^2$.
4. Synthesize signal from thus obtained amplitude spectrogram $\boldsymbol{V}$ and stored phase spectrogram $\boldsymbol{\phi}$ by applying inverse STFT to $\boldsymbol{V}e^{\sqrt{-1}\boldsymbol{\phi}}$.

The third processing is the most essential in many signal processing techniques on spectrogram, while other processings are rather routine. The most important point is how to distinguish the singing voice component from others in the third stage above. Majority of the state-of-the-art singing voice extraction techniques considered to extract singing voice on a time-frequency domain utilizing some properties on singing voice such as timbrel features, high-rankness, harmonicity, etc, as follows.

Exploiting Timbre of Singing Voice

Ozerov et al. [114], [115] focused on the difference of spectral distribution (timbre) of singing voice and instruments, and modeled them by Gaussian mixture model (GMM). In their method, the GMM was trained in advance in a supervised way, and tuned adaptively for each input.

Harmonicity-based Approach

Lagrange et al. [85] utilized a technique of computer vision so as to pick up the harmonically related spectral peaks of the short-time spectra of singing voice.

Pitch Estimation and Time-frequency Masking

Some studies utilized the pitch information of singing voice. In Li and Wang's method [92], segments including singing voice were first detected based on spectral features. Then, at each of the detected singing-voice segments, the predominant pitch was estimated with using the autocorrelation and thresholding.

Hsu and Jang [62] extended this approach to enable to capture unvoiced components of the singing voice with utilizing the spectral envelope information.

NMF-based Approaches

Another popular stream is based on NMF of music spectrogram [138], where it is assumed that spectrogram of music can be expressed as an assemblage of a limited number of spectral templates.

In Vembu and Baumann's method [151], spectral templates obtained by NMF were classified into singing voice and others with their spectral features such as Mel-frequency Cepstral Coefficients (MFCC), LFPC, and PLP.

Virtanen et al. [160] utilized NMF with pitch inference. In their method, the pitch of singing voice was first estimated based on multiple F_0 estimation technique [130], then, the singing voice was roughly removed based on the pitch, and the residual was used for training accompaniment model with NMF. Finally, the singing voice was extracted from the mixture using the derived accompaniment model.

High Rankness of Singing Voice

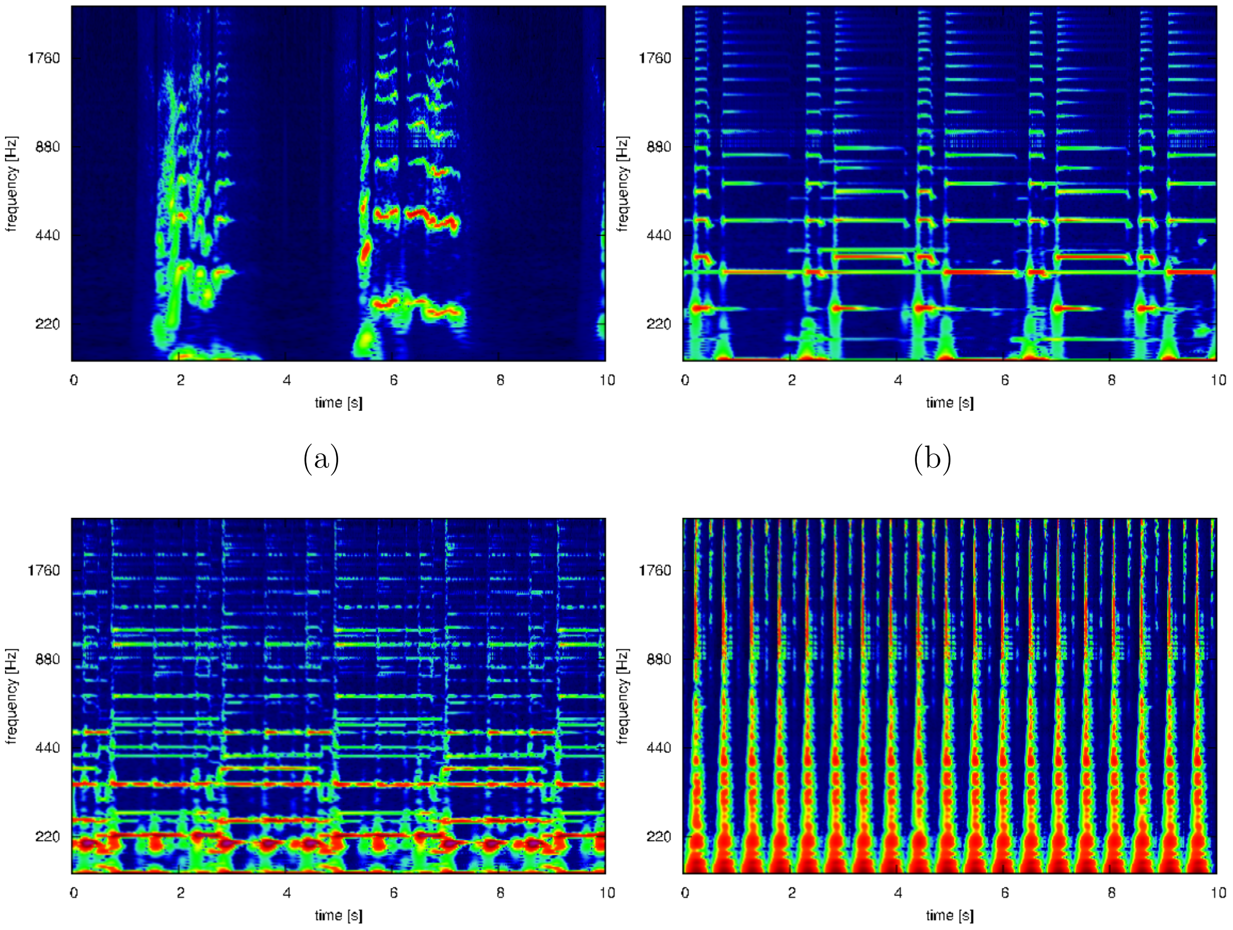
Some of other studies also have focused on low-rankness of music spectrogram. Huang et al. [67] assumed that the spectrogram of accompaniment would lie in lowrank subspace while singing voice would not, as accompaniments are rather repetitive while singing voice are less so. They then proposed a technique based on principal component analysis (PCA). Rafii and Pardo [120] proposed a method "REPET" that suppressed repeating components in spectrogram, i.e. accompaniments. Some of NMF-based approaches may also be classified into this class. Raj et al. [121] modeled an NMF-like generative signal model and applied probabilistic inferences.

2.2 Fluctuation-based Singing Voice Enhancement

2.2.1 Observation of the Spectrograms of Music Signals

Let us see some spectrograms[⋆1] of real music signals. Figure 2.1 shows examples of the spectrograms of music signals. In these spectrograms, we may observe following

[⋆1] From here, we show both log-frequency and linear-frequency spectrograms. Log-frequency spectrogram (sometimes called as scalogram) is obtained by Constant Q Transform (CQT), while linear-frequency spectrogram are obtained by STFT in this thesis. Note, for visibility of lower frequencies, we showed log-frequency spectrogram here, but all the processings of this chapter are carried out on spectrograms (STFT spectrogram), because of the efficiency and simplicity of STFT.



(a)

(b)

(c)

(d)

Figure 2.1:: Examples of the log-frequency spectrogram of music signals. All are excerpted from "dreams" in BASS-dB database [154]; (a) vocal, (b) bass, (c) guitar, (d) drums.

1. Singing voice has fluctuation. Figure 2.1 (a)

2. Singing voice has harmonic structure. Figure 2.1 (a)

3. Instruments such as bass, guitar, have harmonic structure. Figure 2.1 (b), (c)

4. Instruments such as bass, guitar, are sustained for a while. Figure 2.1 (b), (c)

5. Drums basically occupy wide frequencies instantly. Figure 2.1 (d)

In addition to the discussion above, singing voice has some other notable properties, which may not be evident, and we did not pointed out above; an important properties of singing voice is its characteristic spectral envelope, i.e., the power ratios of the harmonics, which is known to be an important factor of the "timbre" of sounds, and has been used in many studies on singing voice extraction, etc.

It is a little surprising that many of state-of-the-art techniques, which are reviewed in the previous section, did not exploit the salient nature of singing voice, i.e., fluctuation, but exploited other less visible property i.e. timbre, less discriminative property i.e. harmonicity, and "less intuitive" technical property i.e. the rank of the subspace, though we may interpret that the last one implicitly exploited the fluctuation of singing voice.

A reason why fluctuation has not been explicitly exploited, in our hypothesis, is the difficulties to model them on spectrogram, because of their complicated shapes and versatility; that is, their shapes on spectrogram is not fixed, but has variety, as shown in Figure 2.1. Because of the complexity of the spectral shapes, as well as their versatility, it may have been difficult to formalize these properties.

Nonetheless, from the observations above, it seems quite natural to try to exploit the fluctuation of singing voice, because of the obvious visibility on spectrograms. Moreover, another motivation to exploit fluctuation is its "orthogonal" property to other feature of singing voice namely harmonics and timbre; we may suppose that joint use of these characteristics may improve the performance of vocal extraction and other relevant techniques. Thus the study of fluctuation-based signal separation technique is of importance. In the rest of this chapter, we shall consider a technique of singing voice processing that is based on fluctuation.

2.2.2 Possible Approaches to Fluctuation

Parametric model-based or Not

A possible approach to the problem is the use of parametric model and the parameter inference. That is, it is common practice to formulate a fluctuating signal as

$$x(t) = \sum_i a_i(t) \sin \left(\int^t \phi_i(t') dt' \right). \tag{2.1}$$

where $a_i(t), \phi_i(t)$ are certain functions. There have been some studies as such, but there are some drawbacks in this approach as follows. Firstly, if the model was so "complete," or in other words, if the model have too many freedoms, then it would be quite difficult to determine all the parameters in the model. To the contrary, if the model was so "simple," it would be controversial whether such a simple model really represent the real-world singing voice.

Because of the difficulties above, we avoid this approach in this dissertation, but consider other approaches that are not based on any specific signal models. The approaches that are based on modulation spectrum, described below, may be classified into this class.

Modulation Spectrum

Modulation spectrum [91], [141], RASTA [59] are basically the frequency analysis of the envelope of the signal, which may be roughly interpreted as an analysis of the frequency distribution of AM (amplitude modulation). Modulation spectrum was principally developed in speech processing area, because of the importance of 1–10 Hz fluctuation of speech signals in ASR applications, and some speech techniques exploiting these modulations are proposed [58], [73], [74], [80], [147].

In music area, there exist some technique that exploit the fluctuation of singing voice [2], [84], [96], [97], [122], [126], though they are basically discrimination and detection tasks; despite the existence of the plenty of application to recognition tasks, however, the studies that explicitly used the modulation of the signals in source separation problems are limited. The exceptions includes the studies carried out by the school of Les Atlas [2], [135], etc. The study by Les Atlas and Janssen [2] aimed at separating a music signal based on modulation spectrum, though the example shown in their paper [2] was not on singing voice enhancement, but only a toy example of the separation of the sounds of a flute and a castanet; it may require further studies to apply the technique to singing voice enhancement in real-world recordings.

Nonetheless, modulation spectrum may not be the only way to capture the fluctuation of signals, but we may consider other approaches on the based on other ideas. The approach of this thesis, indeed, would be one that is promising.

2.3 Intermediate Nature of Singing Voice between 'Harmonic' and 'Percussive'

This section characterizes the singing voice comparing to other typical components in music, namely piano, guitar and percussion. We then classify the musical sounds into three components, and place singing voice as an "intermediate component" between "harmonic" and "percussive."

Three Typical Components:

↑ **Sinusoidal components**

$\mathcal{H}$ Sinusoidal, flatly-played instruments, (e.g., piano, guitar),

$\mathcal{V}$ Fluctuated quasi-stationary component (e.g., singing voice),

$\mathcal{P}$ Transient, non-stationary instruments, (e.g., percussion).

↓ **Instantaneous non-stationary components**

Figure 2.2:: Three typical components in terms of the depth of fluctuation.

2.3.1 From the Perspective of the Depth of Fluctuation

Before considering the singing voice, let us first consider the cases of a piano, for reference. Due to its mechanical structure, the pitch of piano tones and its harmonics basically do not change, or only slightly change if any, in a single note. We can also apply the same discussion to some other instruments such as a guitar, though it has more frequent exceptions (e.g., pitch bend) than a piano. As just described, the sound of the pitched instruments such as pianos and guitars basically do not have fluctuation, or have slight fluctuation if any.

A singing voice, unlike those instruments, typically has fluctuation of pitch and amplitude. Since the vocal cord is a human organ which is not as stable as artifacts, it does not generate sounds as flatly as the instruments above do, by and large. Besides the mechanical constraints, many singers fluctuate their singing voice for musical and emotional expressions. This fluctuation is called vibrato, which is another reason that a singing voice has much fluctuation than the instruments.

Because of the reasons above, we can assume that a singing voice is “less sinusoidal” than the instruments such as a piano and a guitar in many cases. At the same time, a singing voice is obviously “much more sinusoidal” than percussions. To summarize those two facts, we can regard singing voice as “intermediately sinusoidal,” as well as “intermediately non-sinusoidal” sound, between sustained instruments and percussive instruments. In other words, we can consider three classes as described in Figure 2.2, which are typified by piano, singing voice, and percussions.

2.3.2 Another Perspective: "Thriftiness" of Time-frequency Expansion

Let us describe the situation above in another language: "thriftiness" of the time-frequency expansion, which focuses on how "economically" a signal is represented by the time-frequency bases. In this section, we describe the idea in symmetric two ways.

Thriftiness of Time-frequency Expression in Frequency

In order to represent a $\boldsymbol{\mathcal{H}}$ component such as piano on a time-frequency representation, we need only a few bases; $\boldsymbol{\mathcal{H}}$ is approximated as

$$\sum_{n=1}^{\nu} a_n \sin nk_0(t + t_0 + \phi_n) \tag{2.2}$$

around a time t_0, and the signal can be expressed by only ν bases,

$$\left\{\exp(2\pi\sqrt{-1}k(t - t_0)/L); k = k_0, 2k_0, \ldots, \nu k_0\right\}, \tag{2.3}$$

which is the subset of all the DFT bases (i.e. $\nu \ll K$),

$$\left\{\exp(2\pi\sqrt{-1}k(t - t_0)/L); k = 0, 1, \ldots, K\right\}, \tag{2.4}$$

where $K := L/2 + 1$ in all.[⋆2] In other words, $\boldsymbol{\mathcal{H}}$ is "thrifty" in frequency, or more precisely, the distribution of $\boldsymbol{\mathcal{H}}$ is limited around the harmonics $k = k_0, 2k_0, \ldots, \nu k_0$, and the signal occupies only a few frequency bins. (See Figure 2.1 (b), (c).)

To the contrary, in order to represent a $\boldsymbol{\mathcal{P}}$ component, we need many bases, because it is approximately white on the timing of the beats. (See Figure 2.1 (d).) That is, we need instantaneously many bases to represent percussive sounds on spectrogram; the number of bases required here may be almost all the K bases, because of its instantaneous whiteness. In other words, $\boldsymbol{\mathcal{P}}$ is not thrifty in frequency.

The question is how many bases are needed to express a $\boldsymbol{\mathcal{V}}$ component instantaneously. Our answer to the question is shown below, in section 2.5, but we intuitively answer here that "it may be between ν and K," because fluctuating components need more bases than sinusoidal signals in general, since AM and FM expand the bandwidth, while it may not need as many bases as percussions. We may summarize the discussion as shown in Figure 2.3.

[⋆2] Note we assumed k_0 integer, which can be a real number in general. We ignored window function here. Both for make discussion simpler.

Three Typical Components:

↑ **Thrifty in Frequency** (Requires less time-frequency bases)

$\boldsymbol{\mathcal{H}}$

$\boldsymbol{\mathcal{V}}$ **? (to be discussed in section 2.5)**

$\boldsymbol{\mathcal{P}}$

↓ **Not Thrifty in Frequency** (Requires more time-frequency bases)

Figure 2.3:: Three typical components in terms of the "thriftiness" of time-frequency representation, in frequency.

Three Typical Components:

↑ **Not Thrifty in Time** (Requires more time-frequency bases)

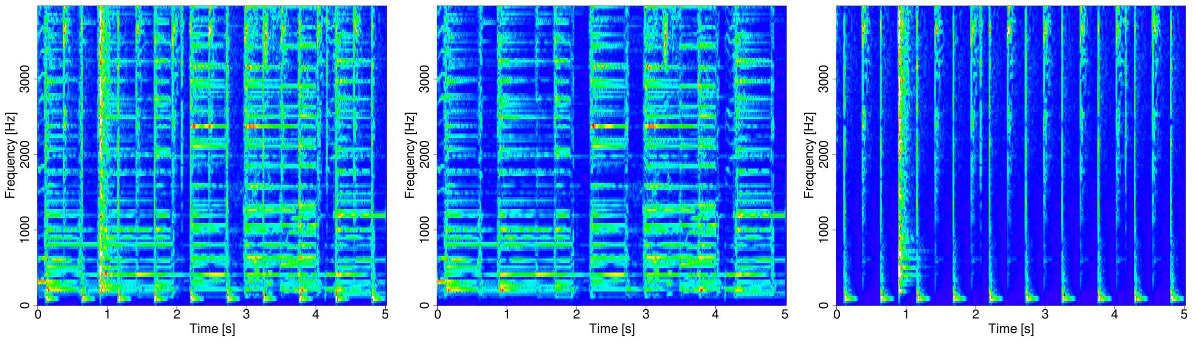
$\boldsymbol{\mathcal{H}}$

$\boldsymbol{\mathcal{V}}$ **? (to be discussed in section 2.5)**

$\boldsymbol{\mathcal{P}}$

↓ **Thrifty in Time** (Requires less time-frequency bases)

Figure 2.4:: Three typical components in terms of the "thriftiness" of time-frequency representation, in time.



(a) (b) (c)

Figure 2.5:: Spectrograms (linear-frequency contrary to Figure 2.1) of (a) a mixed music signal $\boldsymbol{Y}$ (extracted from a song "dreams"), (b) a harmonic component $\boldsymbol{H}$, in which it is observed that it is rather continuous in time than in frequency, and (c) a percussive component $\boldsymbol{P}$, which is continuous in frequency.

Thriftiness of Time-frequency Expression in Time

We may similarly consider the “thriftiness” of time-frequency expansion in the direction of time, symmetrically. That is, focusing on a certain frequency k_0, $\boldsymbol{\mathcal{P}}$ components requires only a few component, because it is temporally thrifty, while $\boldsymbol{\mathcal{H}}$ requires many bases

$$\left\{\exp(2\pi\sqrt{-1}k_0(t-t')/L); t' = 0, s, 2s, \ldots, \tau s\right\}. \tag{2.5}$$

because it is not thrifty temporally. $\boldsymbol{\mathcal{V}}$ also requires discussion in this case, which shall be discussed in section 2.5. We may summarize the discussion as shown in Figure 2.4.

2.4 Harmonic/Percussive Separation Ignoring Vocal

In the previous section, we classified a music signals into three classes, $\boldsymbol{\mathcal{H}}$, $\boldsymbol{\mathcal{V}}$, and $\boldsymbol{\mathcal{P}}$, and discussed as follows;

$\boldsymbol{\mathcal{H}}$ is smooth in time, because it is sustained for a while, while it does not occupy many frequency bins. In other words, it is thrifty (or localized) in frequency, around the harmonics, while they are not thrifty temporally. It seems as if horizontal lines on spectrograms as shown in Figure 2.5 (b).

$\boldsymbol{\mathcal{P}}$ is smooth in frequency, because it is impulse-like. In other words, it is not thrifty in frequency, while it is thrifty around the times of the beats. It seems as if vertical lines on spectrograms as shown in Figure 2.5 (c).

$\boldsymbol{\mathcal{V}}$ may behave intermediately between $\boldsymbol{\mathcal{H}}$ and $\boldsymbol{\mathcal{P}}$, but we postpone the discussion to section 2.5.

Tentatively, let us forget the existence of $\boldsymbol{\mathcal{V}}$ which has a “complicated” property, and consider a simpler problem of separating $\boldsymbol{\mathcal{H}}$ and $\boldsymbol{\mathcal{P}}$ components, in this section. The separation of $\boldsymbol{\mathcal{V}}$ from the mixed music signals shall be worked out in section 2.5, on the basis of the technique discussed in this section.

2.4.1 A Simplest Horizontal/Vertical Extraction Technique and its Drawback

In our purpose to decompose a spectrogram into $\boldsymbol{\mathcal{H}}$ (horizontally continuous as if it is a horizontal line) and $\boldsymbol{\mathcal{P}}$ (vertically continuous as if it is a vertical line), a simplest way is just applying line detection techniques, edge detection techniques, etc., of computer vision, such as Sobel filter. However, we may not use this technique directly, but we had better to consider another technique based on the similar idea but extended, which is consistent with some requirements: “reconstructivity” and “non-negativity.” Reconstructivity means that the sum of separated two signals, harmonic $h(t)$ and percussive $p(t)$, should be almost identical to the original signal $y(t)$, i.e., $y(t) \approx h(t)+p(t)$, because what we intend to do here is source decomposition. Non-negativity is required because each element of spectrogram is the “power,” which cannot be negative.

2.4.2 Brief Introduction of HPSS

Motivation to HPSS

Because of the reasons above, we need to consider another “horizontal/vertical separation technique” taking these requirements into account. In summary, we may decompose a spectrogram $\boldsymbol{Y} \approx \boldsymbol{H}+\boldsymbol{P}$ into $\boldsymbol{H}$ and $\boldsymbol{P}$ by solving problems that satisfy following requirements.

1. The technique separate a signal into “horizontal” and “vertical” components, focusing on the “derivative,” similarly to some of the basic filers in computer vision.
2. The sum of separated spectrograms should be approximately identical to the original spectrogram
3. The separated spectrograms should be non-negative.

This kind of problem, actually, has already been addressed by Miyamoto, Ono, et al. in [100], [101], [112], [113]. The series of techniques are called HPSS, and there have already been some applications [111] based on HPSS including audio chord estimation [125], [149] and rhythm map generation [145].

In this dissertation, we shall use this method as a fundamental technique, because it complies with our purpose here, and it already has some successful applications. In the rest of this subsection, we briefly review HPSS. For more details, see Chapter 6.

Formulation

The method separates a signal $y(t)$ on spectrogram domain on the basis of three assumptions. The first assumption is that the spectrograms of $h(t)$ and $p(t)$, i.e.,

$$\boldsymbol{H} = (H_{n,k})_{(n,k)} = |\mathsf{STFT}(L)\,[h(t)]\,|^2 \in \mathbb{R}^{N\times K} \tag{2.6}$$

$$\boldsymbol{P} = (P_{n,k})_{(n,k)} = |\mathsf{STFT}(L)\,[p(t)]\,|^2 \in \mathbb{R}^{N\times K} \tag{2.7}$$

are "smooth" in time and in frequency, respectively, The assumption reflects the nature of harmonic components and percussive components. That is, the harmonic components are rather "smooth" in time, because they are sustained for a while, while percussive components are rather "smooth" in frequency, because they are instantaneous. On the basis of the assumption, Ono et al. defined criteria to measure how strongly the assumptions are satisfied as follows,

$$S_{\mathrm{H}}(\boldsymbol{H}) := \frac{1}{2\sigma_{\mathrm{H}}^2}\sum_{n=1}^{N-1}\sum_{k=0}^{K-1}(H_{n,k}^{\gamma} - H_{n-1,k}^{\gamma})^2, \tag{2.8}$$

$$S_{\mathrm{P}}(\boldsymbol{P}) := \frac{1}{2\sigma_{\mathrm{P}}^2}\sum_{n=0}^{N-1}\sum_{k=1}^{K-1}(P_{n,k}^{\gamma} - P_{n,k-1}^{\gamma})^2, \tag{2.9}$$

where σ_{H} and σ_{P} are weighting parameters, which were defined empirically as 0.3 in [112], and γ is an exponential factor.

The second assumption is that the sum of the separated signals should be almost equal to the original signal, i.e., $y(t) = h(t) + p(t)$. On power spectrogram, it is approximated as

$$\boldsymbol{Y} \approx \boldsymbol{H} + \boldsymbol{P}. \tag{2.10}$$

Quantitatively, the approximation error can be measured by some divergence, such as Itakura-Saito divergence$^{\star 3}$. In HPSS, in order to evaluated the proximity of $\boldsymbol{H}+\boldsymbol{P}$ to $\boldsymbol{Y}$, Ono et al. exploited the generalized Kullbuck-Leibler divergence $D_{\mathrm{KL}}(\boldsymbol{Y}^{2\gamma} \parallel \boldsymbol{H}^{2\gamma} + \boldsymbol{P}^{2\gamma})$, which is known to be a "good" criterion in the literature of source separation such as NMF. The explicit form of generalized KL divergence shall be defined in Chapter 6. The third assumption is simply the non-negativity, $\boldsymbol{H} \geq 0, \boldsymbol{P} \geq 0$

Algorithm 2.6 HPSS updating formulae

```
procedure HPSS($\boldsymbol{H}, \boldsymbol{P}, \boldsymbol{m}; \boldsymbol{Y}$)
    $a_1 \leftarrow 2(1+\sigma_{\mathrm{H}}^{-2})$
    $a_2 \leftarrow 2(1+\sigma_{\mathrm{P}}^{-2})$
    for $\forall(n,k)$ do
        $b_1 \leftarrow \sigma_{\mathrm{H}}^{-2}\,({H_{n-1,k}}^{\gamma} + {H_{n+1,k}}^{\gamma})$
        $b_2 \leftarrow \sigma_{\mathrm{P}}^{-2}\,({P_{n,k-1}}^{\gamma} + {P_{n,k+1}}^{\gamma})$
        $c_1 \leftarrow 2m_{n,k}{Y_{n,k}}^{2\gamma}$
        $c_2 \leftarrow 2\,(1-m_{n,k})\,{Y_{n,k}}^{2\gamma}$
        ${H_{n,k}}^{\gamma} \leftarrow (b_1 + \sqrt{b_1^2 + 4a_1c_1})/2a_1$
        ${P_{n,k}}^{\gamma} \leftarrow (b_2 + \sqrt{b_2^2 + 4a_2c_2})/2a_2$
        $m_{n,k} \leftarrow {H_{n,k}}^{2\gamma}({H_{n,k}}^{2\gamma} + {P_{n,k}}^{2\gamma})$
    end for
end procedure
```

.

On the basis of these assumptions above, HPSS was formulated as an optimization problem to find the optimal spectrograms $\boldsymbol{H}$ and $\boldsymbol{P}$ that minimize the following objective function.

$$\begin{aligned} \text{minimize } & U(\boldsymbol{H}^{\gamma}, \boldsymbol{P}^{\gamma} | \boldsymbol{Y}^{\gamma}) := S_{\mathrm{H}}(\boldsymbol{H}^{\gamma}) + S_{\mathrm{P}}(\boldsymbol{P}^{\gamma}) \\ & \quad + D_{\mathrm{KL}}(\boldsymbol{Y}^{2\gamma} \| \boldsymbol{H}^{2\gamma} + \boldsymbol{P}^{2\gamma}) \\ \text{subject to } & \forall(n,k), H_{n,k} \geq 0, P_{n,k} \geq 0. \end{aligned} \tag{2.11}$$

[⋆3] Assuming that the complex spectorgrams $\hat{H}_{n,k}$ and $\hat{P}_{n,k}$ follow the complex normal distribution, i.e., real and imaginary parts of $\hat{H}_{n,k}$ follow $\mathcal{N}(0, H_{n,k}^2/2)$, and those of $\hat{P}_{n,k}$ follow $\mathcal{N}(0, P_{n,k}^2/2)$, respectively, and assuming that they are independent, then $\hat{Y}_{n,k}$ (complex spectrogram of $y(t) = h(t) + p(t)$) also follows the complex normal distribution, i.e., real and imaginary parts follow $\mathcal{N}(0, (H_{n,k}^2 + P_{n,k}^2)/2)$. Then its squared amplitude $Y_{n,k}^2$ follows exponential distribution,

$$P(Y_{n,k}^2 | H_{n,k}^2, P_{n,k}^2) = \sigma^{-2}\exp(-Y_{n,k}^2/2\sigma^2), \text{ where } \sigma^2 = (H_{n,k}^2 + P_{n,k}^2)/2.$$

The log likelihood function is

$$\begin{aligned} \sum_{n,k} \log P(Y_{n,k} | H_{n,k}^2, P_{n,k}^2) &= \sum_{n,k} \left(-\frac{Y_{n,k}^2}{H_{n,k}^2 + P_{n,k}^2} + \log \frac{2}{H_{n,k}^2 + P_{n,k}^2}. \right) \\ &= -D_{\mathrm{IS}}(Y_{n,k}^2 | H_{n,k}^2 + P_{n,k}^2) + \sum_{n,k} \log Y_{n,k} + \text{const}, \end{aligned}$$

where $D_{\mathrm{IS}}(x|y) = \sum\{(x/y) - \log(x/y) - 1\}$. Assuming that $\sum_{n,k} \log Y_{n,k}$ is a constant, it is concluded that the minimization of Itakura-Saito divergence implies the maximization of log likelihood. Once we verified that we may discuss the error between $\boldsymbol{Y}^2$ and $\boldsymbol{H}^2 + \boldsymbol{P}^2$ by a kind of divergence, we may “generalize” the discussion to other kind of distance measure. In case of the original HPSS in [112], generalized KL divergence was exploited.

By solving the problem, we can separate a spectrogram $\boldsymbol{Y}$ into two spectrograms $\boldsymbol{H}, \boldsymbol{P}$; algorithm 2.6 shows the explicit procedure to obtain $\boldsymbol{H}$ and $\boldsymbol{P}$. This is followed by Wiener filtering and inverse STFT to synthesize audible waveforms $h(t)$ and $p(t)$ as describes above. Hereinafter, let us simplify the notation of the whole procedure as follows,

$$y(t) \xrightarrow{\mathsf{HPSS}(L)} h(t), p(t). \tag{2.12}$$

The computation of each update is efficient, and the solution rapidly converges to near the optimal value within a small number of iterations.

Advantages of HPSS

One of the principal advantages of this approach is the simplicity and locality. HPSS does not require the information of the bins which are much distant from the bin (n, k) under consideration, similarly to Sobel filter above. This locality forms the advantage in real-world applications practically. Thus the computation complexity is not much different from the simplest Sobel filter, except that we consider the iterations of HPSS updating formula, because it is guaranteed in HPSS that the iteration monotonically improve the objective function, without expiring the constraints.

Other advantages are that it is unsupervised method which requires no pre-training, and it does not require any prior knowledge of input music signal; it needs not to know what kind of instruments are included in the song.

Moreover, the HPSS algorithm is executable in real time, by the procedure shown in Algorithm 4.3 in page 57. In real time processing, instead of applying updating formula of Algorithm 2.6 for several times, a sliding block of size I is used as shown in line 8–11, Algorithm 4.3. Practically, setting I around 10^1 to 10^2, a solution with a sufficient quality is obtained. The issues on sliding block analysis shall be discussed later again in Chapter 4.

2.5 Singing Voice Enhancement Based on Two-stage HPSS

In this section, we discuss the nature of a singing voice and consider how to extract it from music audio signals. On the basis of the discussion, we show the key idea of our singing voice extraction method, which is one of the most important contribution of this dissertation.

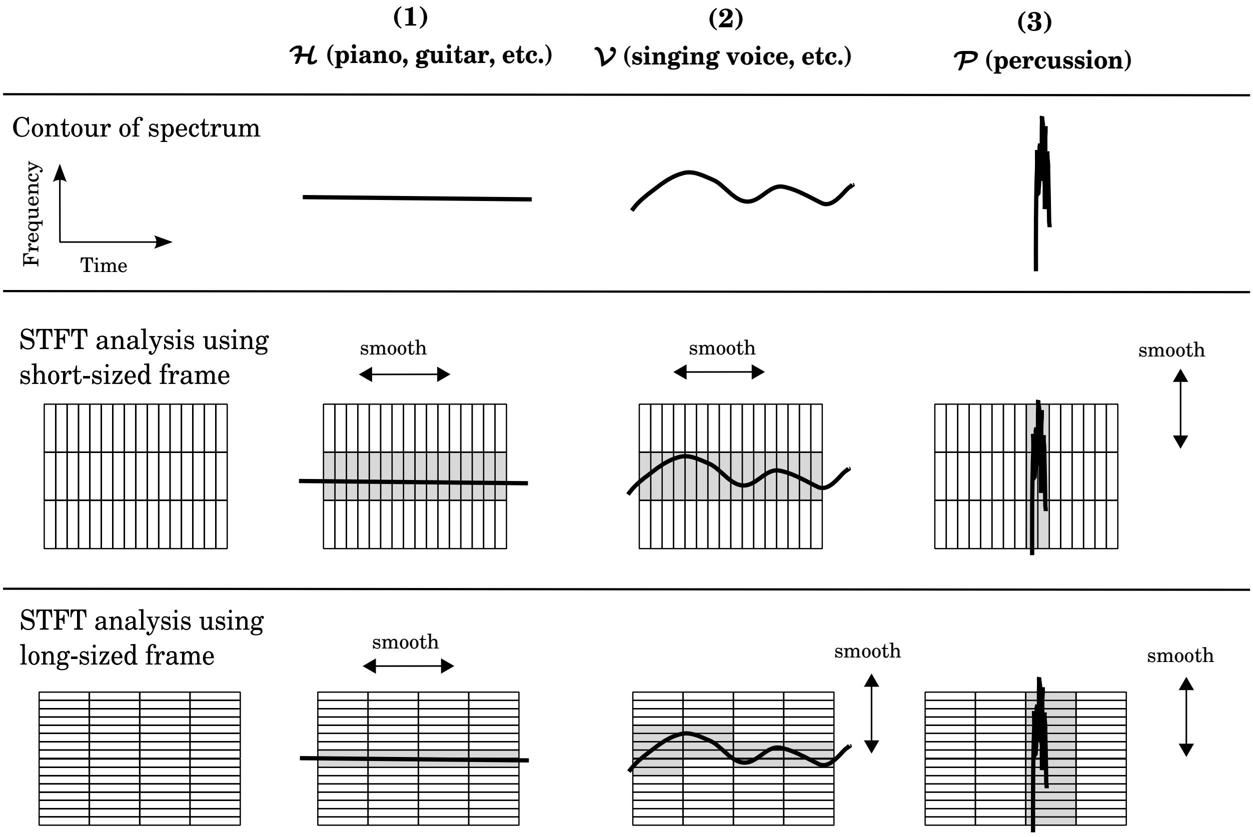

(1)
$\mathcal{H}$ (piano, guitar, etc.)
(2)
$\mathcal{V}$ (singing voice, etc.)
(3)
$\mathcal{P}$ (percussion)
Contour of spectrum
Frequency
Time
STFT analysis using short-sized frame
smooth
smooth
smooth
STFT analysis using long-sized frame
smooth
smooth
smooth

Figure 2.7:: The dependencies of the appearance of the spectrogram of three types of signals, on each spectrogram. (1) $\mathcal{H}$ component is smooth in time, nonsmooth in frequency, on both short- and long-framed STFT domains. (2) $\mathcal{V}$ component is smooth in time, nonsmooth in frequency, on short-framed STFT domain. However, on long-framed STFT domain, it is nonsmooth in time compared to the width of the time-frequency bins, and smooth in frequency compared to the height of the time-frequency bins. (Note that these figures are not necessarily the exact illustration of the effects of pitch fluctuation, but intuitive ones. Another point to note is that, not only the pitch fluctuation (frequency modulation) but also the amplitude fluctuation (amplitude modulation) expand the bandwidth of the spectrum.) (3) $\mathcal{P}$ component is almost always nonsmooth in time, and smooth in frequency, regardless of the frame length.

2.5.1 Dependencies on Time/frequency resolution (or the Bases Set)

Let us consider how to extract intermediate component $\mathcal{V}$ with fluctuation from mixed audio signals, which was ignored in HPSS. Our idea is the utilization of two different time-frequency resolution, or in other words, two different bases set, or two different frame lengths L. In following subsections let us see the details on the dependencies of the behavior of $\boldsymbol{\mathcal{V}}$ on the frame length L.

2.5.2 $\boldsymbol{\mathcal{V}}$ as a "Harmonic" (Horizontal) Component

First, let us consider a case in which the frame length of STFT is 10 [ms] (i.e., $L = 0.01 \times f_{\mathrm{s}}$), and frame shift s is its half. In this case, the frequency resolution of STFT is 50 Hz, because the product of temporal and frequency resolution is $s/L = 1/2$. Because of its poor frequency resolution, small fluctuation of $\mathcal{V}$ falls in only a few of frequency bins, while the signal occupies many temporal bins because its pitch does not change within such a short duration. Therefore, its appearance on $\mathsf{STFT}[L = 0.01 \text{ [s]}]$ is quite similar to $\mathcal{H}$ (middle row of Figure 2.7) in terms of the local smoothness.

We may also interpret the situation as follows. The temporal scale $L = 10$ [ms] is too short comparing to the temporal-scale of fluctuation of singing voice, which would be around 100 [ms]. In this short temporal-scale, the fluctuation of "much longer time-scale" can be ignored, and is regarded to be a "constant." Thus $\mathcal{V}$ appears quite similar to $\mathcal{H}$. In other words, these time-frequency bases efficiently decomposes $\mathcal{V}$ in frequency, while not thrifty in time, similarly to $\mathcal{H}$.

For these reasons above, when we apply HPSS to the spectrogram whose frame length is L_1 which is short enough, a signal $s(t)$ is roughly separated into $\mathcal{H} + \mathcal{V}$ and $\mathcal{P}$ as follows,

<u>*HPSS (short frame)*:</u>

$$s(t) \xrightarrow{\mathsf{HPSS}(L_1)} h_1(t),\ p_1(t), \tag{2.13}$$

$$\text{where } \ h_1(t) \approx \mathcal{H} + \mathcal{V}, \quad p_1(t) \approx \mathcal{P}. \tag{2.14}$$

Thus we can remove the $\mathcal{P}$ component from the mixed music signals.

2.5.3 $\mathcal{V}$ as a "Percussive" (Vertical) Component

Next, we have to decompose $h_1(t)$ into a harmonic component $\mathcal{H}$ and a singing voice $\mathcal{V}$. In order to achieve that, let us consider a case in which the frame length of STFT is 1 [s]. In this case, the frequency resolution of the spectrogram is 0.5 Hz. In contrast to the previous case, fluctuation of $\mathcal{V}$ is much broader than the frequency resolution, and it occupies many frequency bins in a single frame. In other words, the fluctuation of the singing voice is compressed into a few number of time frames, and the nature appears not as "fluctuation" but as "broadbandness" on this spectrogram.

We may also interpret the situation as follows. The temporal scale $L = 1$ [s] is too long comparing to the temporal-scale of fluctuation of singing voice; around 100 [ms]. The effect of the fluctuation cannot be ignored in this long temporal-scale, and $\mathcal{V}$ appears quite dissimilar to $\mathcal{H}$, but rather similar to $\mathcal{P}$ in this macroscopic perspective. In other words, these time-frequency bases do not efficiently decomposes $\mathcal{V}$, as efficiently as $\mathcal{H}$ in frequency, but it consumes more bases similarly to $\mathcal{P}$.

Thus, the appearance of $\mathcal{V}$ component on the spectrogram is not similar to $\mathcal{H}$ but to $\mathcal{P}$, as shown in the bottom row of Figure 2.7. Consequently, using a sufficiently long analyzing frame L_2, we can separate $h_1(t)$ into the following two components by HPSS,

HPSS (long frame):

$$h_1(t) \xrightarrow{\mathsf{HPSS}(L_2)} h_2(t),\ p_2(t), \tag{2.15}$$

$$\text{where}\quad h_2(t) \approx \mathcal{H}, \quad p_2(t) \approx \mathcal{V}. \tag{2.16}$$

The obtained $p_2(t)$ roughly corresponds to the $\mathcal{V}$ component, which is the target component.

2.5.4 Whole the Procedure and Experimental Example of Two-stage HPSS

In summary, applying HPSS twice on differently-resolved two spectrograms separates a music signal into three components, $\mathcal{H}$, $\mathcal{V}$ and $\mathcal{P}$ as shown in Figure 2.8, and thus obtained $p_2(t)$ would roughly be the singing voice, which we aimed at in this chapter.

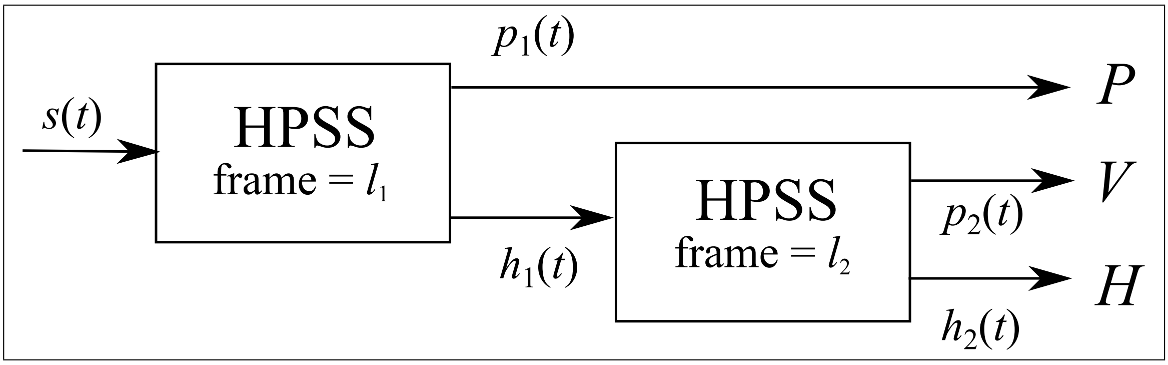


Figure 2.8:: Diagram of two-stage HPSS.

In order to verify the effectiveness of the singing voice enhancement based on two-stage HPSS, we conducted an experiment on singing voice enhancement using a real-world music audio signal. The music signals we used for experiments were excerpted from the RWC music database [46]. The data were resampled to 16 kHz and converted into monaural signals by adding both channels of stereo signals.

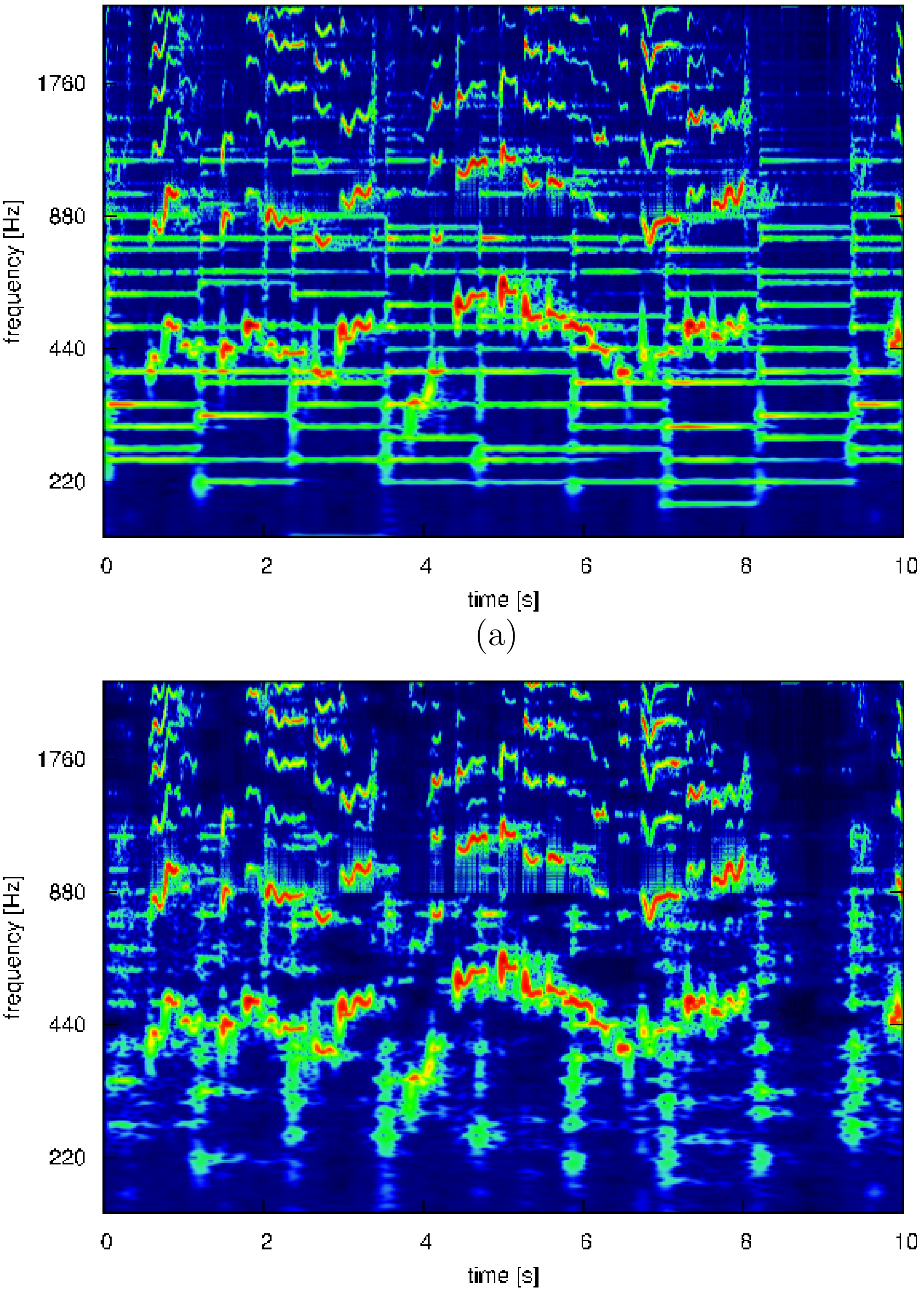
Figure 2.9 shows a result of two-stage HPSS. The figures show the spectrograms of a input signal (Figure 2.9 (a)) and the $\mathcal{V}$ component extracted by two-stage HPSS (Figure 2.9 (b)). We can see in Figure 2.9 (b) that most accompanying sounds are suppressed effectively by the method, and the singing voice is clearer in spectrogram (b) than that of the spectrogram (a). The figures show that the method effectively extracts the singing voice from the mixed music audio signal.

2.6 Large Scale Evaluation on Two-stage HPSS

2.6.1 Experimental Condition

To verify the effectiveness of the two-stage HPSS, we conducted experiments on the singing voice enhancement using music audio signals. The criteria for the performance evaluation of singing voice enhancement were the Normalized SDR (NSDR) and the Generalized NSDR (GNSDR). NSDR is defined as the improvement of SDR as follows,

$$\mathrm{NSDR}[x^{\mathsf{estim}}(t); s(t), x^{\mathsf{GT}}(t)] = \mathrm{SDR}[x^{\mathsf{estim}}(t); x^{\mathsf{GT}}(t)] - \mathrm{SDR}[s(t); x^{\mathsf{GT}}(t)], \tag{2.17}$$



(a)

(b)

Figure 2.9:: An Example of the result. (a) The log-frequency (CQT) spectrogram of an input signal (10 seconds from RWC-MDB-P-2001, No. 25 [46]), (b) the result of two-stage HPSS.

where $x^{\mathsf{estim}}(t), s(t)$ and $x^{\mathsf{GT}}(t)$ denote the estimated signal, the input signal, and the target signal, respectively. SDR [51][⋆4] is defined by

$$\mathrm{SDR}[x(t); y(t)] = 10 \log_{10} \left(\langle x, y \rangle^2 \Big/ \left(\|x\|^2 \|y\|^2 - \langle x, y \rangle^2 \right) \right). \tag{2.18}$$

GNSDR is defined as the averaged NSDR of all the pieces, weighted by w_i, the length of i-th piece [115],

$$\mathrm{GNSDR} = \sum_i w_i \mathrm{NSDR}[\hat{x}_i(t), s_i(t), x_i(t)] \Big/ \sum_i w_i. \tag{2.19}$$

Those criteria were also used in some previous works [62], [67], [115], [120].

For evaluation dataset, we exploited the MIR-1K database [62][⋆5], which is comprised of 1000 Chinese songs sung by amateur singers. The same dataset was used in some of previous works [62], [120]. The length T of each clip was around 4 to 13 [s]. All the data were monaural, and the sample rate of them was f_{s} =16 kHz. The vocal part and the accompaniment part were recorded separately, and we could mix them in any signal to noise ratio (SNR) (signal to noise ratio, i.e., voice to accompaniment ratio). In this study, we mixed the singing voice and the accompaniment in $-10, -5, 0, 5$, and 10 dB for the experiment.

The parameters we used were as follows. The frame shift s_i was half the length of the frame length L_i. The length of the frames were $L_1/f_{\mathrm{s}} = 8$ [ms] ($L_1 = 128$ points) and $L_2/f_{\mathrm{s}} = 512$ [ms] ($L_2 = 8192$ points). Both analyzing window $g_1(t)$ and reconstructing window $g_2(t)$ were sine window, $g_1(t) = g_2(t) = \sin \pi t/l, (0 \le t < l)$. Under this condition, the following equation is satisfied for any signal $x(t)$,

$$x(t) \equiv \mathsf{STFT}^{-1}(L)[\mathsf{STFT}(L)\,[x(t)]]. \tag{2.20}$$

The parameters of HPSS were as follows. The updating scheme was based on the sliding block analysis, which shall be shown in Chapter 4. The size of the sliding block (≈ number of iteration) I was 30. The values of $\sigma_{\mathrm{H}}, \sigma_{\mathrm{P}}$ were 0.3. These values were identical to those which were described in the Ono's first HPSS paper [112]. After two-stage HPSS, we applied

[⋆4] SDR is sometimes defined by $\mathrm{SDR}[x(t); y(t)] = 10 \log_{10}(\|ay(t)\|^2 / \|n(t)\|)$, where $x(t) = ay(t) + n(t)$, and $n(t)$ is noise which is perpendicular to the signal, i.e., $\langle y, n \rangle = 0$. Under the assumption, a is uniquely decided as $a = \langle x, y \rangle / \|y\|^2$, and the following formula immediately follows,

$$\begin{aligned} a^2 \|y(t)\|^2 / \|x(t) - ay(t)\|^2 &= a^2 \|y\|^2 / (\|x\|^2 - 2a\langle x, y \rangle + a^2 \|y\|^2) \\ &= (\langle x, y \rangle^2 / \|y\|^2) / (\|x\|^2 - 2\langle x, y \rangle^2 / \|y\|^2 + \langle x, y \rangle^2 / \|y\|^2). \end{aligned}$$

from which the "definition" above (2.18) is derived.

[⋆5] [Online (Visited Oct. 2013)] `http://unvoicedsoundseparation.googlepages.com/mir-1k` high pass filter to cut off any components lower than 110 Hz, because vocal components are less likely to appear in such lower frequencies.

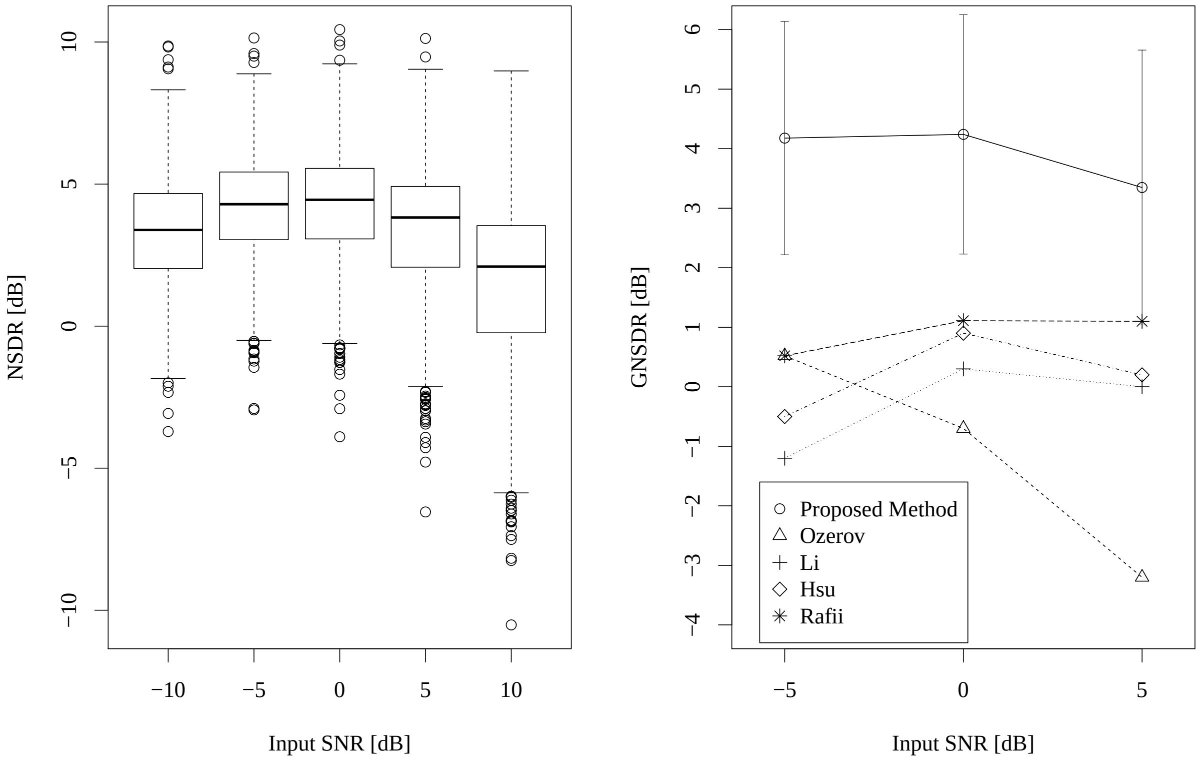
2.6.2 Results and Discussion

Fig 2.10 (a) shows the distribution of NSDR for each input SNR condition. For most samples in most SNR conditions, NSDR were larger than 0 dB, i.e., SDR were improved. The method performed well, especially in -5 dB and 0 dB conditions. Fig 2.10 (b) compares GNSDR between the proposed method and some existing methods, namely Ozerov et al. [114], Li and Wang [92], Hsu et al. [62] and Rafii and Padro [120]. The figure shows that the proposed method considerably outperformed other methods in any SNR conditions within -5 to 5 dB.

The method was especially effective when the music signal sufficed the assumptions. That is, if the singing voice had sufficient fluctuation, and if it was accompanied by very stationary component and very percussive component, the performance tended to be better. However, the method was not typically effective in the following two cases. One was when the singing voice was not fluctuating sufficiently. When the singer singed flatly, or the singing voice was sustained for a long while with slight fluctuation, the singing voice did not satisfy the assumption that "singing voice has fluctuation," and two-stage HPSS did not separate the component into $\mathcal{V}$ components, but into $\mathcal{H}$ components. The other was when accompanying sounds were fluctuating. Typical instruments were violin, trumpet, etc., which fluctuate to some extent. Those sounds tended to be separated into $\mathcal{V}$ components, because they satisfy the assumption of singing voice, i.e., fluctuation. To remove those sounds, we have to use other properties of sounds such as timbre, but this is outside of the scope of this thesis.

2.7 Summary of Chapter 2

In this chapter, we described two-stage HPSS, a singing voice enhancement method in monaural music signals. The method extracts the singing voice in music signals focusing on its fluctuation. Such natures of singing voice are exposed on two differently-resolved spectrograms, one is obtained using a short frame (around 10 [ms]), and the other is obtained using a long frame (around 500 [ms]). On the former spectrogram, both $\mathcal{H}$ (sustained, such as a piano) component and $\mathcal{V}$ (quasi-stationary, fluctuating, such as a singing voice)



(a) (b)

Figure 2.10:: (a) Boxplot of NSDR of the proposed method for 1000 songs in MIR-1K dataset. (b) GNSDR comparison with some existing singing voice enhancement techniques. The perpendicular bars on the plot of our method indicate the weighted standard deviation of NSDR. GNSDR of existing methods were cited from [62] and [120]. All the GNSDR scores are calculated using 1000 songs in MIR-1K dataset.

component appear as a “smooth-in-time” components, while $\mathcal{P}$ (transient, such as a percussive instrument) component appears as a “smooth-in-frequency” component. On the latter spectrogram, however, the behavior of $\mathcal{V}$ component is not similar to that of $\mathcal{H}$, but is similar to $\mathcal{P}$ component, i.e., it appears as a “smooth-in-frequency” component. Therefore, two-stage application of HPSS on differently-resolved two spectrograms roughly separates $\mathcal{V}$ component from $\mathcal{H}$ and $\mathcal{P}$ components.

We evaluated the method in the same framework as some previous works. According to experimental evaluations, the performance of the proposed method was considerably higher than those of some previous works and marked around 4 dB GNSDR for -5, 0, and 5 dB mixtures.

The result of this chapter shall be utilized following two chapters; Chapter 3 and 4. In Chapter 3, we will use the method as a preprocessing for melody estimation in audio signals. In Chapter 4, we will utilize the residual accompanying signals in an automatic karaoke generator. The concept of this chapter is extended in Chapter 5, in which we consider using more than two spectrograms.

Chapter 3

Application of Two-stage HPSS to Singing-melody F_0 Estimation in Mixed Music Signals

3.1 Introduction

Extracting the information on melody, especially the sequence of the frequency, from music signals, is one of the most basic Music Information Retrieval (MIR) problems, because of its familiarity to the listeners, as well as the affluence of the information it has. As discussed in Chapter 1, humans can easily recognize which component is the melody in a music signal, which is indeed another reason we study the melody extraction technique; i.e., this problem would be one of the "focal points" to reduce the gap between the "ears" of humans and computers.

Some difficulties of melody pitch estimation comes from the existence of the accompanying instruments in music signals; in music, many sounds are played simultaneously, and synchronously, and furthermore, accompanying sounds always have many harmonics, and the harmonics sometimes share the same frequency with those of the melody. Nonetheless, let us recall that the problems above have been partly solved effectively by using the fluctuation of singing voice in Chapter 2, if melody is played by singing voice. Indeed, as we may find

Notes on Chapter 3: Some parts of this chapter is the revision of the authors' papers [302],[305]. Note there is another technique that exploited the idea of two-stage HPSS as a preprocessing for melody extraction, which was proposed by Hsu et al. [65], [66].

Note, this section sometimes uses some fundamental knowledge of music. Readers who are unfamiliar with the terminologies of music, see Appendix A.

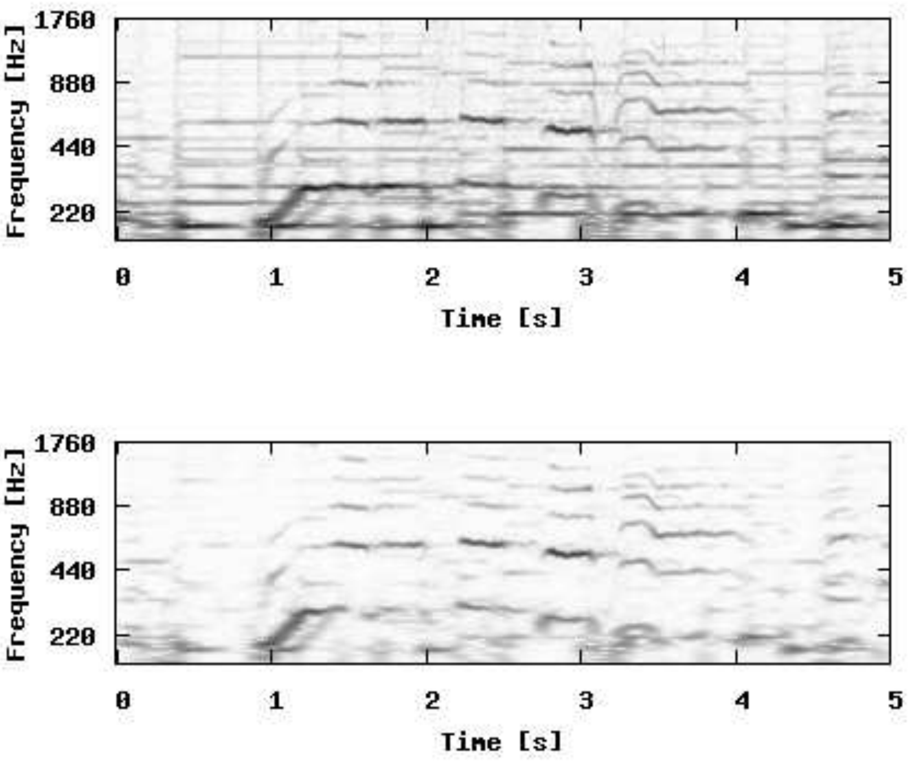


Figure 3.1:: Spectrograms of original signal (excerpted 5[s] from train06.wav, LabROSA dataset) and the signal processed by two-stage HPSS.

in Figure 3.1, two-stage HPSS can effectively suppress the accompanying instrument, and it can be expected that we may rather easily estimate the pitch sequence from the bottom figure of Figure 3.1. Thus it would be natural to consider that applying the technique as a preprocessing is helpful in our aim to estimate the pitch sequence of melody from the mixed music signal.

In this chapter, instead of considering the general problem of audio *melody* extraction from mixed music signals, we consider a specific problem of audio *singing-melody* extraction, which would be a natural application of the two-stage HPSS discussed in the previous chapter. This problem would be a reasonable subproblem of the general melody extraction problems, because the melodies are often played by singing voice in many real-world musics. In this chapter, we first apply two-stage HPSS in order to roughly extract singing voice from the mixed music signals. This processing is followed by an F_0 tracking technique, which is supposed to be very simple owing to the preprocessing. The results of this chapter prove the effectiveness of the two-stage HPSS as a preprocessing for an MIR technique.

3.1.1 Related Work

There are many studies that focused on the predominant pitch estimation from the mixed music signals. One of the earliest works that addressed the task was PreFEst [47], which was based on a probabilistic model. Besides PreFEst, several melody tracking algorithms have since been proposed, e.g., the methods by Fujihara et al. [43], Cao et al. [14], Durrieu et al. [25], [27], Salamon and Gómez [134]. Some of the other singing pitch transcription methods, e.g., Ryynänen and Klapuri's method [130], V. Rao and P. Rao's method [124], are formulated as a subproblem of a general multiple F_0 estimation, which is also an important topic in the music signal processing area [81], [82].

3.2 Problem Definition in AME MIREX

Because of the importance of the melody extraction from music signals, an exchange on Audio Melody Extraction (AME) has been held as a part of Music Information Retrieval Exchange (MIREX) [22][⋆1] every year since 2005, when MIREX began. To the date, many participants have submitted their algorithms to the exchange, including those who are listed in related work.

In this dissertation, we adopt the rule used in AME branch in MIREX, since the rule of AME MIREX may be regarded as one of the standard rules of this task for now, because it has been a leading event in this field in the last decade. Thus, we consider formulating our method and conducting the experiments in line with the rules of AME MIREX. This section describes the rules of AME MIREX.

3.2.1 Input/Output Data Format in AME MIREX

In MIREX, participants are supposed to submit a program to the committee, and the committee evaluates the performance of each program. Each program is given audio signals $x(t), (t \text{ integer such that } 0 \le t < f_{\mathrm{s}}T)$, and what it is supposed to estimate is the sequence of instant frequency

$$(x_n \text{ [Hz]})_{0 \le n < N} = (x_0 \text{ [Hz]}, x_1 \text{ [Hz]}, \cdots, x_{N-1} \text{ [Hz]}), \tag{3.1}$$

[⋆1] [Online (Visited Oct. 2013)]
`http://www.music-ir.org/mirex/wiki/2009:audio_melody_extraction_results`, etc.

output_format_example.dat:

n	time ($n\Delta$) [s]	frequency (x_n)[Hz]
0	0.00	263.0
1	0.01	263.0
2	0.02	263.2
3	0.03	263.1
4	0.04	440.0
5	0.05	440.2
6	0.06	439.8

Figure 3.2:: Example of AME MIREX output format

of the target signal (melody), around the time $n\Delta$[s], where $n(0 \leq n < N := T/\Delta)$ denotes the index of time frame, $\Delta = 10$ [ms] is the temporal resolution, $f_{\mathrm{s}} = 16000$ Hz is the sampling rate, and T [s] is the duration of an input song.

The programs are supposed to output thus estimated sequence of the instant frequency $(x_n \text{ [Hz]})_{0 \leq n < N}$ by a format as shown in Figure 3.2. Note, if there is no melody in n-th frame (i.e., at the instant $n\Delta$ [s]), it is defined that $x_n = 0$ [Hz], but we do not consider the task of melody absence estimation in this dissertation.

3.2.2 Evaluation Criteria in AME MIREX

MIREX evaluators have the ground truth, $(x_n^{\mathsf{GT}}\text{[Hz]})_{0 \leq n < N}$

$$(x_n^{\mathsf{GT}} \text{ [Hz]})_{0 \leq n < N} = (x_0^{\mathsf{GT}} \text{ [Hz]}, x_1^{\mathsf{GT}} \text{ [Hz]}, \cdots, x_{N-1}^{\mathsf{GT}} \text{ [Hz]}), \tag{3.2}$$

which were labeled by hand, by the creators of the dataset. The evaluators evaluate the difference between the estimated (x_n) and the ground truth (x_n^{GT}), then they present the result. The evaluation criteria to measure the correctness are following 5 criteria, which are defined in [117].

1. *Voicing Recall*
2. *Voicing False Alarm*
3. *Raw Pitch Accuracy* (RPA)

4. *Raw Chroma Accuracy*
5. *Overall Accuracy*

Overall Accuracy is the most general criterion on this task, which is defined as follows,

$$(\text{Overall Accuracy}) = \frac{\#(\text{correctly estimated frames})}{N}. \tag{3.3}$$

This value indicates the general performance of this task. That is, the criterion includes the performance of melody absence detection and therefore, this criterion is not suitable in our purpose, then we ignored in this dissertation.

Instead of that, we especially focused on RPA, which is defined as the ratio of correctly estimated segments in melody-active segments.

$$\text{RPA} = \frac{\#(\text{correctly estimated frames in melody-active frames})}{\#(\text{all melody-active frames})} \tag{3.4}$$

In other words, this criterion ignores the errors in melody-absent frames.

The "correctness" of the estimated pitch for each segment is judged by whether the difference between the estimation and the ground truth is within a half semitone (50 cents) or not [117]. That is[⋆2],

$$\text{“}x_n \text{ is correctly estimated”} \;:\Leftrightarrow -50 < 1200 \log_2 \left(\frac{x_n \text{ [Hz]}}{x_n^{\mathsf{GT}} \text{ [Hz]}} \right) < 50. \tag{3.5}$$

In other words, if the estimated frequency is not distant from the ground truth more than half semitone, it is regarded that the frequency is correctly estimated.

3.3 Simple Pitch Tracking Algorithm

3.3.1 Overview of the Formulation of the Algorithm

This section describes a simple technique of estimating a pitch sequence from an audio signal, that is applied after two-stage HPSS. We do not claim the novelty of the method, but what is most important here is that the method is quite simple.

[⋆2] Note, $1200 \log_2$ (Frequency ratio) indicates the interval between two frequencies in cent.

Constant Q Transform of the Given Signal

Given a signal, we first apply two-stage HPSS to it. Thus we obtain a signal $s(t)$, in which the sounds of accompanying instruments are suppressed. Let $\boldsymbol{S}$ be a CQT of the signal $s(t)$,

$$\boldsymbol{S} = (S(n,x))_{0\leq n<N, X_{\mathsf{L}}\leq x\leq \mathsf{Nyq}} \tag{3.6}$$

$$= (\boldsymbol{S}_0, \boldsymbol{S}_1, \ldots, \boldsymbol{S}_{N-1}), \tag{3.7}$$

$$\text{where } \boldsymbol{S}_i = (S(i,x))_{X_{\mathsf{L}}\leq x\leq \mathsf{Nyq}}, \tag{3.8}$$

where $N(:= T/\Delta)$ denotes the number of time frames of the song. We write $S(i,x)$ as $S_i(x)$ for readability. X_{L} is the minimum MIDI note number we consider. Nyq is the MIDI note number of Nyquist frequency, i.e., $f_{\mathrm{s}}/2 = 440 \times 2^{(\mathsf{Nyq}-69)/12}$.

Requirement for the Solution

Using thus obtained CQT spectrogram $\boldsymbol{S}$, we estimate the sequence of MIDI note number $x_i \in \mathbb{R}$,

$$\boldsymbol{x} = (x_i)_{0\leq i<N} = (x_0, x_1, \ldots, x_{N-1})., \quad \text{where} \quad X_{\mathsf{L}} \leq x_n \leq X_{\mathsf{H}}. \tag{3.9}$$

where X_{H} is the highest candidate of the pitch x_i. Note we do not need to consider the pitches higher than X_{H}, because there is a limitation of the height of the voice that humans can utter.

In order to estimate the sequence of MIDI note number $\boldsymbol{x}$, we considered an optimization problem. In designing the objective function, we considered following two simple assumptions. We shall define explicit forms of these assumptions, and integrate them into an optimization problem in the next subsection.

Pitch–Spectrum Correlation: A pitch x_i should be consistent with the observed spectrum at the same instant $\boldsymbol{S}_i$.

Pitch Continuity: A pitch x_i should be close to the previous pitch x_{i-1}.

Discretization

In computation, we naturally need to discretize MIDI note number. A natural way of discretization is $x_n = m_n/R + X_{\mathsf{L}}$, where $m_n \in \mathbb{N}$, and $R \in \mathbb{N}, (R \geq 1)$. R denotes the number of frequency bins within a semitone. For example, when $R = 10$, the frequency resolution of CQT is 10 cents (= 1/10 semitone.) The choice of R depends on the trade-off between frequency resolution and the computation efficiency.

3.3.2 Explicit Formulation

Requirement 1: Spectrum–Pitch Consistency (Correlation):

In order to evaluate the correlation between the observed short-time spectrum $\boldsymbol{S}_i$ and the pitch x_i, we define a criterion as follows. We first consider the correlation between the CQT spectrum $\boldsymbol{S}_i$ and a sound model $q(\xi)$, as follows,

$$\begin{aligned}\mathsf{Correlation}(x_n, \boldsymbol{S}_n) &= \int_{x_n}^{\mathsf{Nyq}} S_n(\xi) q(\xi - x_n) d\xi, &(3.10)\\ &= \sum_{m=0}^{(\mathsf{Nyq}-x_n)R} S_n\left(x_n + \frac{m}{R}\right) q\left(\frac{m}{R}\right) \frac{1}{R}, &(3.11)\end{aligned}$$

where we discretized the integral as follows,

$$\int_{X_1}^{X_2} f(x)dx := \sum_{m=0}^{(X_2-X_1)R} f\left(X_1 + \frac{m}{R}\right) \frac{1}{R}. \tag{3.12}$$

The sound model $q(\xi)$ is defined by an artificial spectrum, whose β-th harmonic has $1/\beta$ amplitude of the F_0. That is,

$$q(\xi) = \begin{cases} \dfrac{1}{[\beta]} & \text{if } \beta := 2^{\xi/12} \text{ is approximately a positive integer} \\ 0 & \text{otherwise} \end{cases} \tag{3.13}$$

where $[\beta]$ denotes the nearest integer to β. To be specific, the values of $q(\xi)$ are defined as follows[⋆3],

$$\begin{aligned} &q(0) = 1, \quad q(12) = \frac{1}{2}, \quad q(19) = \frac{1}{3}, \quad q(24) = \frac{1}{4}, \\ &\qquad q(28) = \frac{1}{5}, \quad q(31) = \frac{1}{6}, \quad q(34) = \frac{1}{7}, \quad q(36) = \frac{1}{8}, \\ &\qquad\qquad q(38) = \frac{1}{9}, \quad \ldots, \quad q(\text{otherwise}) = 0. \end{aligned} \tag{3.14}$$

[⋆3] Note that $2^0 = 1,,$ $2^{12/12} = 2,$ $2^{19/12} \approx 2.9966,$ $2^{24/12} = 4,$ $2^{28/12} \approx 5.0397,$ $2^{31/12} \approx 5.9932,$ $2^{34/12} \approx 7.1272,$ $2^{36/12} = 8,$ $2^{38/12} \approx 8.9797, \ldots$.See also Appendix C.2. The digits $0, 12, 19, 24, 28, \ldots$ correspond to the following musical intervals, which are quite frequently used in many European-origin music; 0 semitone = unison, 12 semitones = 1 octave, 19 semitones = 1 octave + perfect 5th, 24 semitones = 2 octaves, 28 semitones = 2 octaves + major 3rd, etc. This is actually a reason for the difficulties of music signal processing. That is, there is an ambiguity whether a spectral component is a fundamental frequency of a musical note, or just a harmonics of another musical note. For example, it is not trivially specified whether a spectral component around 440 Hz is the fundamental component of a musical note A4 (69 in MIDI note number), 2nd harmonics of A3 (57), 3rd harmonics of D3 (50), or 5th harmonics of F2 (41) etc. What makes the problem more difficult is that these notes, i.e. F2, D3, A3, A4, are often played simultaneously. The combination of these notes is a typical example of “D minor chord,” and the “minor chord” is one of the most “ordinary” chords in ordinary musics.

It is natural to assume that thus defined $\mathsf{Correlation}(x_n, \boldsymbol{S}_n)$ should take large value if x_n was the true F_0 of this frame. Then, we directly defined a term of objective function by $\mathsf{Correlation}(x_n, \boldsymbol{S}_n)$. That is, our requirement for the consistency is written as follows.

Requirement 1: Spectrum–Pitch Correlation:

$$\text{Larger} \quad \sum_{n=0}^{N-1} \mathsf{Correlation}(x_n, \boldsymbol{S}_n) \quad \text{is preferable.}$$

Requirement 2: Pitch Continuity:

For pitch transition model, we required the difference between x_{n-1} and x_n to be small, because the pitch do not jump large intervals all of a sudden, but the nearer pitch to the last pitch x_{n-1} is likely to be the subsequent pitch. In order to evaluate that, we defined following function.

$$\mathsf{Discontinuity}(x_{n-1}, x_n) = (x_n - x_{n-1})^2. \tag{3.15}$$

Using this function our requirement is written as follows,

Requirement 2: Pitch Continuity:

$$\text{Smaller} \quad \sum_{n=0}^{N-1} \mathsf{Discontinuity}(x_{n-1}, x_n) \quad \text{is preferable.}$$

Optimization Problem based on the Criteria Above:

By summarizing two requirements that larger $\sum_n \mathsf{Correlation}$ and smaller $\sum_n \mathsf{Discontinuity}$ are preferable, we may simply formulate an optimization problem as follows,

Problem:

$$\begin{aligned} \text{maximize} \quad & \sum_{n=0}^{N-1} b(m_n, m_{n-1}, \boldsymbol{S}_n) \\ \text{where} \quad & b(m_n, m_{n-1}, \boldsymbol{S}_t) := \mathsf{Correlation}\,(x_n; \boldsymbol{S}_n) \\ & \qquad -w\mathsf{Discontinuity}\,(x_{n-1}, x_n) \end{aligned}$$

where w is a weight constant to balance each term, which depends on the loudness of the signal. (Note the dimension of $\mathsf{Correlation}$ is the loudness of input signal, while that of $\mathsf{Discontinuity}$ is squared semitone.) Note m_n is always coordinated with x_n by the formula $m_n = (x_n - X_{\mathrm{L}})R$.

In order to solve this problem we used Viterbi algorithm. See standard textbooks including e.g., [6, §13.2.5], [118, §4.7.1], [129, §15.2.3], etc. The explicit procedure of Viterbi algorithm is shown in Algorithm 3.3.

3.4 Evaluation of the Effect of Two-stage HPSS

We conducted an experiment on melody extraction in order to evaluate the effects of two-stage HPSS as a preprocessing.

Dataset

We exploited a referential dataset of MIREX provided by LabROSA at Columbia University [⋆4]. The dataset contains 13 audio files and ground truth F_0 data for each audio file. Nine of 13 clips were vocal songs, and other 4 clips are instrumental pieces generated by MIDI, which were omitted in this experiment. All the data were monaural, and the sampling rate was 16000 Hz. The duration of each song was around 20–30[s].

Parameter Settings of Two-stage HPSS

The parameter of two-stage HPSS is as follows. The lengths of window functions were 256[ms] (4096 samples) and 32[ms] (512 samples). The parameters of two-stage HPSS was $N' = K' = 1, \sigma_{\mathrm{H}} = \sigma_{\mathrm{P}} = 0.3$.

Parameter Settings of Pitch Tracking Algorithm

In the pitch tracking stage, we used following parameters. The search range of x_n was between $X_{\mathsf{L}} = 52$ (165Hz, E3) and $X_{\mathsf{H}} = 76$ (660 Hz, E5), because the frequency outside of this range is less frequently sung. Indeed, the lowest note of Tenor is about C3 (approximately 130 Hz), and highest note of Alto is about E5 (approximately 660 Hz). The frequency

[⋆4] [Online (Visited Oct. 2013)] `http://labrosa.ee.columbia.edu/projects/melody`

Algorithm 3.3 Melody Tracking Algorithm Based on Viterbi Algorithm

```
Initial Frame
$M = (X_{\mathsf{H}} - X_{\mathsf{L}})R$
for all $m \leftarrow 0, \ldots, M-1$ do
    $\mathsf{AccumCost}(1, m) \leftarrow 0$
    $\mathsf{LnkToPrev}(1, m) \leftarrow 0$                                ▷ Need not to define
end for

Forward Algorithm
for $n \leftarrow 1, \ldots, N-1$ do
    for all $m \leftarrow 0, \ldots, M-1$ do
        $\mathsf{AccumCost}(n, m) \leftarrow \max_{m_{\mathsf{PREV}}}[\mathsf{AccumCost}(n-1, m_{\mathsf{PREV}}) + b(m_n, m_{\mathsf{PREV}}, \boldsymbol{S}_t)]$
        $\mathsf{LnkToPrev}(n, m) \leftarrow \operatorname{argmax}_{m_{\mathsf{PREV}}}[\mathsf{AccumCost}(n-1, m_{\mathsf{PREV}}) + b(m_n, m_{\mathsf{PREV}}, \boldsymbol{S}_t)]$
    end for
end for

Backward Search
$m_{N-1} = \operatorname{argmax}_{m}[\mathsf{AccumCost}(N-1, m)]$
for $n \leftarrow (N-2), (N-3), \ldots, 0$ do
    $m_n \leftarrow \mathsf{LinkToPrev}(n+1, \operatorname{argmax}_{m_{\mathsf{NEXT}}} \mathsf{AccumCost}(n+1, m_{\mathsf{NEXT}}))$
end for
```

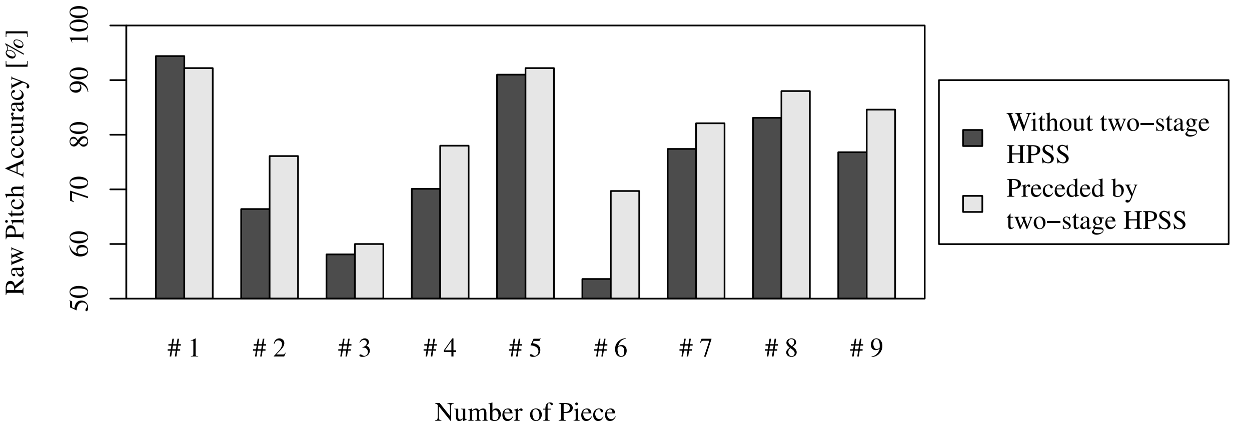


Figure 3.4:: RPA of melody estimation for each piece in LabROSA dataset.

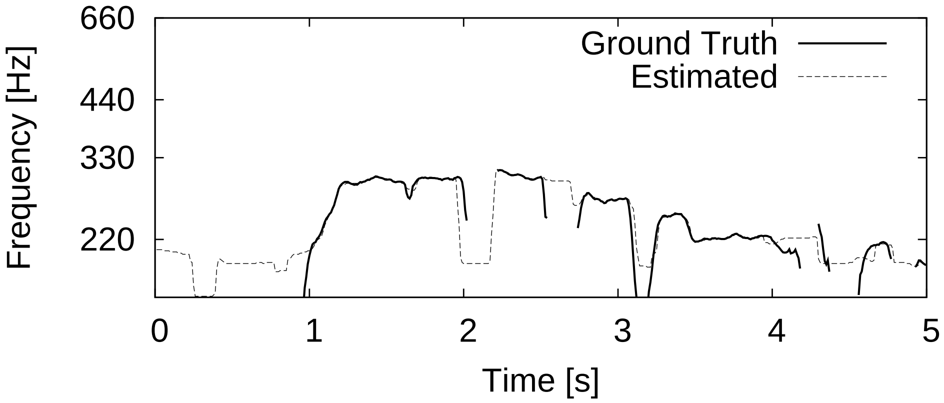


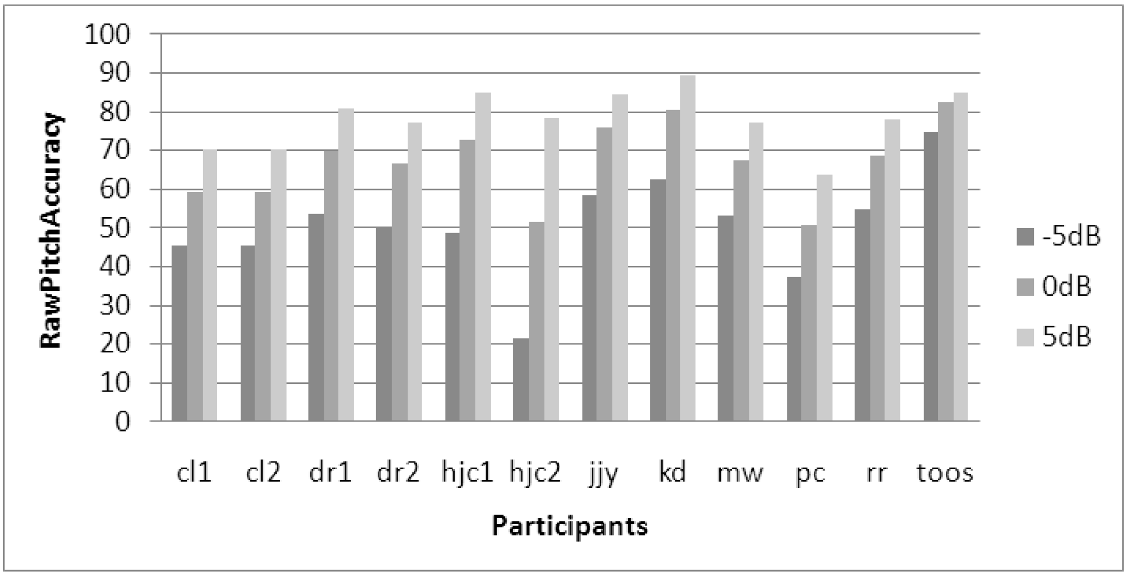
Figure 3.5:: Estimated melody and ground truth of train06.wav (excerpted 10 [s]) in LabROSA dataset.

resolution was 1/10 semitone (10 cents), i.e., $R = 10$. Q value of CQT was 60.0 (approximately equivalent to quarter semitone). The temporal resolution of CQT was 10 [ms]. We set $w = 1/2R\sigma^2$, where $\sigma^2 = 0.2$ [(semitone)^2/ms].

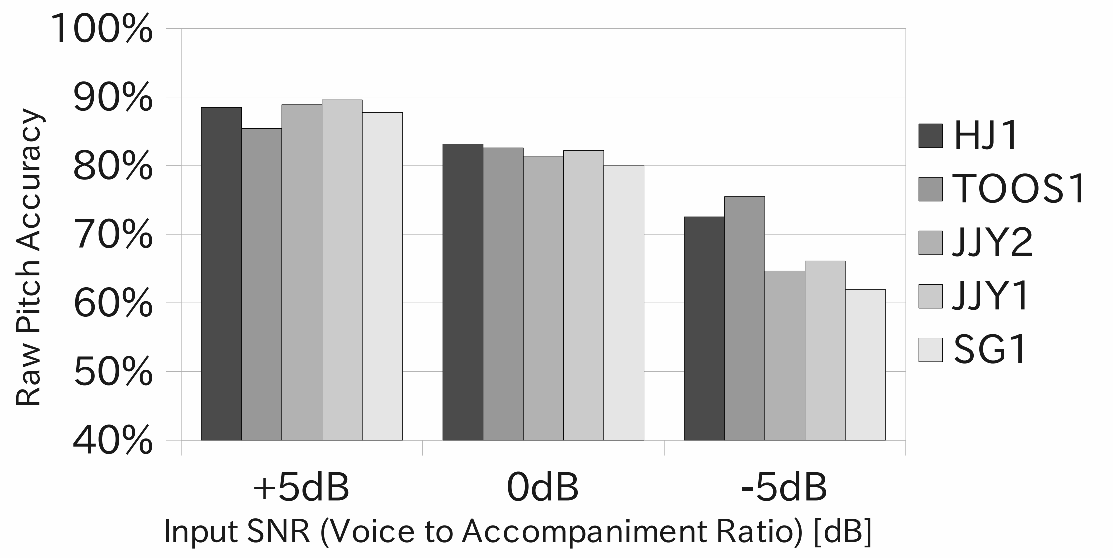
Results and Discussion

Figure 3.4 shows the accuracy ratios for each clip. Compared with the accuracies of the pitch estimation without the preprocessing by two-stage HPSS, it is observed that the two-stage HPSS basically improved the accuracy.

Figure 3.5 shows an example of the result of the tandem connection of two-stage HPSS and pitch estimation, in which it is observed that the pitch is correctly estimated in many frames, except when melody is absent.



(a)



(b)

Figure 3.7:: Excerpted results from AME evaluation, MIREX (a) 2009 and (b) 2010: RPA in +5dB, 0dB, −5dB melody to accompaniment ratio conditions. "TOOS" is our submission. "HJ1" in 2010 also uses a part of our singing voice enhancement technique [305],[316] as a preprocessing. This graph shows that the performance of proposed method is high, especially in low SNR (voice to accompaniment ratio) conditions, and comparable to those of other methods in high SNR (vocal to accompaniment ratio) conditions. It shows the effectiveness of our singing voice enhancement method as a preprocessing for audio melody extraction.

Table 3.6: List of Participants of MIREX AME 2009 and 2010

Participants	2009 ID	2010 ID
Cao et al. [13]	CL1, CL2	–
Durrieu et al. [26]	DR1, DR2	–
Hsu et al. [63, 64]	HJC1, HJC2	HJ1
Joo et al. [71, 72]	JJY	JJY1, JJY2
Dressler [23]	KD	–
Wendelboe [161]	MW	–
Cancela	PC	–
Rao and Rao [123]	RR	–
Salamon and Gomez [133]	–	SG1
Tachibana et al. (proposal)	TOOS	TOOS1

3.5 Comparative Large Scale Evaluation on Audio Melody Extraction in MIREX

3.5.1 Overview of MIREX AME 2009 and 2010

We submitted the method to AME evaluations, held as a part of MIREX 2009 and 2010. In MIREX 2009 AME evaluations, there were 12 submissions from 9 participants, and in MIREX 2010, there were 5 submissions from 4 participants. The participants are shown in Table 3.6

In MIREX evaluations, several datasets were used as listed below, as well as a variety of criteria described above.

ADC04 20 data including songs and the instrumentals

MIREX05 25 data including songs and the instrumentals

INDIAN08 Indian songs

MIREX08 8 songs

MIREX09 (MIR-1K) 374 songs, mixed in -5dB, 0dB, $+5$dB melody to accompaniment ratios.

However, we only focused on the dataset and the criterion that are related to singing voice enhancement. The dataset we focused on was MIR-1K dataset, and the criterion concerned was RPA as discussed previously. MIR-1K datasets of 374 songs, mixed in -5dB, 0dB, $+5$dB

Table 3.8: MIREX 2010 comparison of the run-time of each method. Note the values include some overheads such as launching MATLAB etc.

Method				MIREX09		
	ADC04	MIREX05	INDIAN08	0dB	−5dB	+5dB
HJ1	30m 53s	59m 31s	38m 18s	14h 39m 16s	10h 58m 35s	11h 16m 37s
JJY1	26m 06s	50m 31s	53m 50s	12h 07m 21s	8h 43m 59s	8h 56m 31s
JJY2	36m 17s	1h 09m 30s	52m 14s	14h 06m 20s	14h 10m 22s	13h 55m 51s
SG1	1h 05m 58s	3h 48m 02s	11h 27m 29s	65h 21m 11s	48h 21m 52s	59h 31m 31s
TOOS1	4m 26s	8m 15s	5m 53s	1h 56m 27s	1h 31m 01s	1h 56m 56s

melody to accompaniment ratios. The conditions were similar to those of the experiments in the previous section.

3.5.2 Results of MIREX 2009 and 2010

Figure 3.7 shows the excerpted results from AME evaluation in MIREX 2009[⋆5] and 2010[⋆6]. It shows that the performance of the proposed method (TOOS1) is comparatively high, especially in a condition in which the volume level of singing voice is low compared to accompaniments (−5 dB). The figures also show that the proposed method also performs comparably to other methods in a condition in which the volume level of melody is relatively high (+5 dB). These results show the effectiveness of two-stage HPSS as a preprocessing for singing-melody extraction application.

3.5.3 MIREX 2010 Comparative Evaluation on Computation Efficiency of AME Algorithms

In MIREX, run-time of each program was also evaluated. Table 3.8, which is a summary of the result uploaded on the web site of MIREX[⋆7], shows the run-time of each program submitted to MIREX 2010, though it is not necessarily a fair comparison because of the differences of the languages in which each algorithm is implemented. Nevertheless, it is actually showing that the proposal method required less computation resource than other

[⋆5] [Online (Visited Oct. 2013)] `http://www.music-ir.org/mirex/wiki/2009:mirex2009_results`
[⋆6] [Online (Visited Oct. 2013)] `http://www.music-ir.org/mirex/wiki/2010:mirex2010_results`
[⋆7] [Online (Visited Oct. 2013)] `http://www.music-ir.org/mirex/wiki/2010:Runtime`

programs. This is partly because our program was implemented in C++, which is one of the most efficient languages, but it would be also because the simplicity of the algorithm itself.

Note the results in 2009[⋆8] was not as efficient as the digits described in the table, probably because we had not tuned[⋆9] our CQT program in the date of 2009 submission.

3.6 Summary of Chapter 3

In this chapter, we described a method to extract the pitch series of singing-melody in mixed music signals, which is a straightforward application of two-stage HPSS, described in Chapter 2. An experiment showed that the performance of simple pitch estimation algorithm was improved thanks to the use of two-stage HPSS as a preprocessing. Another experiment showed that the accuracy of the estimation was comparatively high compared with other pitch estimation methods, especially when the input SNR (vocal to accompaniment ratio) is low, due to the proposed singing voice enhancement method. These results also prove the effectiveness of the method in Chapter 2 as a preprocessing for a MIR tasks.

The future work will include constructing a melody-absence model which we excluded in this thesis, by using other features of melody available, e.g., timbrel information and musicological context. Another future work is the investigation on the application other than melody extraction that exploit two-stage HPSS as a preprocessing.

[⋆8] [Online (Visited Oct. 2013)] `http://www.music-ir.org/mirex/wiki/2009:Runtime`

[⋆9] We directly calculated CQT using the definition, i.e., convolution, which is not efficient, as described in Appendix A.

Chapter 4

Audio-to-audio Karaoke System "Euterpe"

4.1 Introduction

Karaoke, which is said to be invented in 1971 [165], can be regarded as one of the early examples of electric-technology-based music applications for amateur music fans. Before long since it was invented, it widely spread and became one of the major leisure especially in Japan, where the popularity of karaoke is comparable to those of watching movies, video games, etc. Indeed, as shown in White Papers [163], [164], karaoke is the 7th populous (38.4 million participants[⋆1]) leisure in Japan in 2011, in number of participants.[⋆2]

On the other hand, it is said that the preference of the music listeners is becoming more diverse recently. Remembering that the current karaoke systems require pre-made MIDI data, which are created manually by professional craftspeople who has an ability of music dictation, it is becoming economically difficult to create all the karaoke data that would cover

Notes on Chapter 4: Some part of this chapter is a collaborative work with Yu Mizuno, a former master student in Sagayama Lab. His contribution is pitch conversion [103], which is described in Appendix C, except the discussion in Appendix C.2.

[⋆1] The population of Japan is about 130 million in 2010.

[⋆2] White Papers [163], [164] claimed that the most populous leisure in Japan was "activities using PC" (71.5 million participants), but it was omitted from this statistics because the research was a web-based one. Excluding PC, most populous leisure in Japan was travelling (55.8 million), followed by dining out (2nd, 53.7 million), going for a drive (3rd, 53.6 million), watching movie (4th, 41.6 million), listening to music (5th, 41.1 million), watching videos (6th, 39.7 million). Karaoke goes beyond other notable leisure include gardening (10th, 33.8 million participants), video games (11th, 33.4 million), card games and board games (12th, 30.9 million), jogging and marathon (17th, 25.9 million), and picnic and hiking (20th, 23.3 million).

such diverse songs, including very old songs, local songs, songs which are appreciated only by a few listeners, and songs by amateur musicians which are distributed through the web sites, etc. Of these, the last one would be our most important target, considering the fact that the recent growth of the web-based music community prompts many amateur musicians to upload their songs, and many listeners enjoy these songs similarly to the songs created by professional musicians. Relevant discussion on the web-based music creation community is found in [54], [55], etc.

In this chapter we show an application of two-stage HPSS described in Chapter 2 to an automatic karaoke generating system, "Euterpe." Since the technique roughly separates a music signal into vocal and instrumental components, the system is applicable to the cases in which separately-recorded tracks of a song nor MIDI data are not available, which were required in the standard karaoke systems. Instead, it only requires the already-mixed music audio signals, such as ordinary CDs, MP3 data, etc. In addition to vocal suppression we also consider pitch transposition, which is another important function of karaoke systems. This technique, similarly to two-stage HPSS, only exploits rather local structure of music signals, and hence, the algorithm is quite efficient, and it works in real-time as a consequence. Thus whole the system works in real-time, which enables the users to modify music signals in real-time.

4.1.1 Related Work

There is a widely-known simple method to generate karaoke signal from mixed music signal, which is simply the subtraction of the channels of stereo signal. That is, it subtracts the right channel $r(t)$ from the left channel $l(t)$ of a stereo signal $(r(t), l(t))$, i.e., $\mathrm{karaoke}(t) = r(t) - l(t)$, (e.g., CenterPanRemover[⋆3], a plug-in for Audacity.) This approach is based on the common practice of present-day stereo recording that all the instruments including singing voice are separately recorded, and the vocal component is placed on the center, when all the parts are mixed down by recording engineers. A drawback of this simplest method is that it can be applied only to these kinds of professionally-created stereo recordings. In other words, the method cannot be applied to live recordings, monaural signals, etc.

In order to cover a wider range of recordings including monaural signals, automatic singing voice removal techniques is required. Because of these backgrounds, a technique that can

[⋆3] [Online (Visited Sep. 2013)] `http://wiki.audacityteam.org/index.php?title=vocal_removal` generate karaoke signals automatically, by erasing singing voice data has significance.

We can consider principally two approaches to the problem. An approach is based on a multiple F_0 analysis and automatic music score estimation, which estimates the music score from the audio signals, and convert it into karaoke MIDI. However, the approach is not easy for now, because there are still many difficulties in multiple F_0 analysis. Another approach is the direct audio-to-audio conversion, similarly to the center pan removal. There have been some studies on automatic karaoke generation for monaural signals, such as [131], using some signal processing techniques. In this chapter, we propose an automatic karoake generation technique based on the latter approach.

4.2 Requirements for the "Euterpe" System

Karaoke is a system which plays vocal-off music as well as guide-melody. The system also has function that tempo and key can be changed by the users interactively, and it shows the lyrics on the display. Of these, we focus on vocal-off generation and key conversion in this dissertation, ignoring guide-addition, tempo conversion[⋆4], and the issues on lyrics. That is, the system requires following properties.

I *Vocal suppression from mixed audio.*

II *Key transposition.* (See Appendix C)

We additionally require the system to satisfy the following property, because of the reason described in the next subsection,

III *Accept Streaming Input*

4.2.1 Remarks on 3rd Requirement: Client-side Real-time Processing

It may not seem that the vocal-off generation needs to work in real-time, because it may seem sufficient practically that a service provider creates the karaoke signals on their servers before they distribute them to the users. However, there is some reasons for the importance

[⋆4] We do not discuss the tempo conversion in this dissertation because it is not necessarily possible in streaming processing. However, it is easily achieved within the same framework when we have whole the audio signal. Indeed, in [103], the tempo conversion is discussed in the same framework assuming the file-input instead of streaming-input.

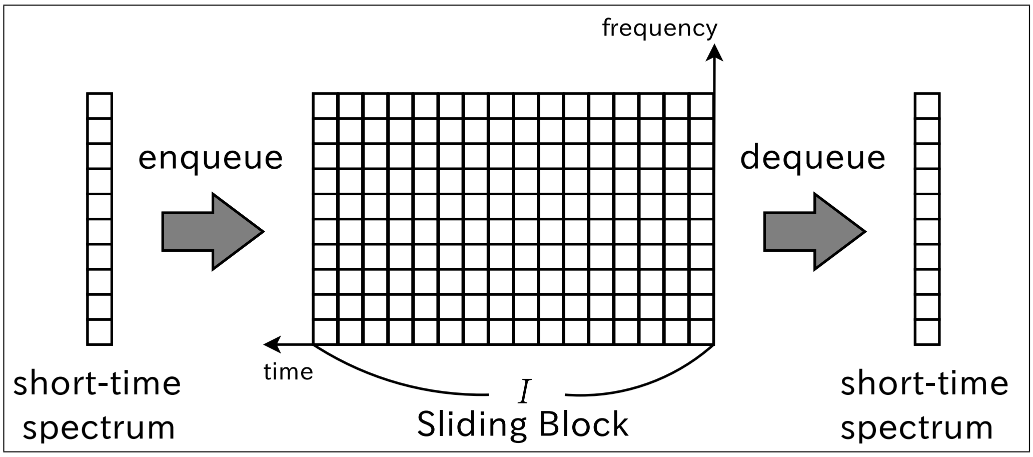


Figure 4.1:: Concept of the sliding block analysis.

to accept streaming input. Let us consider following four possible cases: (Server-side, Client-side) × (Download, Streaming), i.e.,

1. Server-side Vocal-off Generation → Download
2. Server-side Vocal-off Generation → Streaming
3. Download → Client-side Vocal-off Generation
4. Streaming → Client-side Vocal-off Generation

The cases 1, 2, 3, indeed do not require streaming-based vocal-off generation techniques that work in real time. However, when it comes to the case 4, real-time processing is essential.

Although the 4 candidates above have their own advantages and disadvantages, the 4th case is advantageous in a sense that it is user-friendly. In fact, a report by CNN[⋆5] says that downloading has become less common for young listeners in the United States, but they prefer streaming because of the lightheartedness. In addition, client-side signal processing does not require any computation servers, contrary to a server-based service which requires an administrator who deploys the technique in their system, and maintain the server over a long time. Moreover, client-side processing also has an advantage that it is basically independent of the specific services, and the users can use the method on any web sites, theoretically.

[⋆5] [Online (Visited Nov.2013)] `http://edition.cnn.com/2012/06/15/tech/web/music-streaming/index.html`

Algorithm 4.2 HPSS updating formulae for sliding analysis

```
procedure HPSS($\boldsymbol{H}, \boldsymbol{P}, \boldsymbol{m}; \boldsymbol{Y}, i$)
    $a_1 \leftarrow 2(1 + \sigma_{\mathrm{H}}^{-2})$
    $a_2 \leftarrow 2(1 + \sigma_{\mathrm{P}}^{-2})$
    for $\forall(n,k), i \le n < i + I, 0 \le k < K$ do
        $b_1 \leftarrow \sigma_{\mathrm{H}}^{-2}\,({H_{n-1,k}}^{\gamma} + {H_{n+1,k}}^{\gamma})$
        $b_2 \leftarrow \sigma_{\mathrm{P}}^{-2}\,({P_{n,k-1}}^{\gamma} + {P_{n,k+1}}^{\gamma})$
        $c_1 \leftarrow 2m_{n,k}{Y_{n,k}}^{2\gamma}$
        $c_2 \leftarrow 2\,(1 - m_{n,k})\,{Y_{n,k}}^{2\gamma}$
        ${H_{n,k}}^{\gamma} \leftarrow (b_1 + \sqrt{b_1^2 + 4a_1c_1})/2a_1$
        ${P_{n,k}}^{\gamma} \leftarrow (b_2 + \sqrt{b_2^2 + 4a_2c_2})/2a_2$
        $m_{n,k} \leftarrow {H_{n,k}}^{2\gamma}({H_{n,k}}^{2\gamma} + {P_{n,k}}^{2\gamma})$
    end for
end procedure
```

Algorithm 4.3 Whole procedure of Sliding HPSS

```
Preprocessing:
Given an input signal $y(t)$
$\tilde{\boldsymbol{Y}} \leftarrow \mathrm{STFT}_l[y(t)]$                                  ▷ complex spectrogram
$\boldsymbol{Y} \leftarrow |\tilde{\boldsymbol{Y}}|^2$                                     ▷ power spectrogram
$\boldsymbol{H} \leftarrow \boldsymbol{Y}/2$, $\boldsymbol{P} \leftarrow \boldsymbol{Y}/2$            ▷ initial value of $\boldsymbol{H}, \boldsymbol{P}$
$\forall(n,k), m_{n,k} \leftarrow 0.5$                                   ▷ initial value of $\boldsymbol{m}$
HPSS Updating based on Sliding Analysis:
for $-I \le i \le N + I$ do
    for $R_i$ times do
        HPSS ($\boldsymbol{H}, \boldsymbol{P}, \boldsymbol{m}; \boldsymbol{Y}, i$)                        ▷ Algorithm 2.6
    end for
end for
Postprocessing:
$\tilde{\boldsymbol{H}} \leftarrow \sqrt{\boldsymbol{mY}}e^{j\angle\tilde{\boldsymbol{Y}}}, \tilde{\boldsymbol{P}} \leftarrow \sqrt{(\boldsymbol{1}-\boldsymbol{m})\boldsymbol{Y}}e^{j\angle\tilde{\boldsymbol{Y}}}$           ▷ Wiener masking
$h(t) \leftarrow \mathsf{STFT}^{-1}(L)[\tilde{\boldsymbol{H}}], p(t) \leftarrow \mathsf{STFT}^{-1}(L)[\tilde{\boldsymbol{P}}]$     ▷ waveform synthesis
```

4.3 Two-stage HPSS for Singing Voice Suppression

As a singing voice suppressor, we exploited two-stage HPSS discussed in the Chapter 2. Let us recall it briefly. A single HPSS separates a signal into horizontally-continuous (harmonic) components and vertically-continuous (percussive) components by solving an optimization problem on spectrogram domain. In an ordinary condition, HPSS separates a music signal into harmonic components and percussive components. That is,

$$\text{(Music signal)} \xrightarrow{\mathsf{HPSS}(L_1)} \text{(Harmonic + Vocal), (Percussive)} \tag{4.1}$$

where L_1 is a frame length of STFT. To the contrary, if we apply to a spectrogram of long frame length L_2, HPSS separates the signal quite differently. That is, it separates a music signal into components which is sustained for long period (e.g., guitar), and non-stationary components (singing voice and percussive instruments) as follows,

$$\text{(Music signal)} \xrightarrow{\mathsf{HPSS}(L_2)} \text{(Harmonic), (Vocal + Percussive)} \tag{4.2}$$

Specifically, we set the frame length as follows: $L_1 = 512/16000 = 32$ [ms] and $L_2 = 2048/16000 = 128$ [ms]. The overlap and analyzing frames were half the length of the frames, i.e., $s_1 = 16$ [ms] and $s_2 = 64$ [ms] in this chapter.

Then, applying HPSS twice on differently-resolved spectrograms separates a signal into three components, the harmonic (e.g.. guitar), the fluctuated (vocal), and the percussive, as discussed in Chapter 2 as follows,

$$\text{(Music signal)} \xrightarrow{\mathsf{Two\text{-}stage\ HPSS}} \text{(Harmonic), (Vocal), (Percussive)} \tag{4.3}$$

and we may obtain the vocal-off component just by adding the obtained harmonic and percussive components.

Performance of Singing Voice Suppression in terms of SDR

In Chapter 2, we have already shown the performance of the method. The performance of the method as a singing voice “suppression,” actually, is basically the same as the performance of “enhancement,” in terms of SDR, because the following formula holds

$$\mathrm{SDR}[y^{\mathsf{estim}}(t); y^{\mathsf{GT}}(t)] = \mathrm{NSDR}[x^{\mathsf{estim}}(t); s(t), x^{\mathsf{GT}}(t)] \tag{4.4}$$

when

$$x^{\mathsf{estim}}(t) = x^{\mathsf{GT}}(t) + n(t), \quad y^{\mathsf{estim}}(t) = y^{\mathsf{GT}}(t) - n(t) \tag{4.5}$$

$$x^{\mathsf{GT}}(t) + y^{\mathsf{GT}}(t) = x^{\mathsf{estim}}(t) + y^{\mathsf{estim}}(t) = s(t) \tag{4.6}$$

$$\langle x, y\rangle = \langle x, n\rangle = \langle y, n\rangle = 0. \tag{4.7}$$

The proof is an easy direct calculation.

In Chapter 2, Figure 2.10 in page 36, we have verified that

$$\mathrm{NSDR}[(\mathbf{Vocal})^{\mathsf{estim}}; (\mathrm{Mixture}), (\mathbf{Vocal})^{\mathsf{GT}}] \tag{4.8}$$

is approximately 4 [dB]. Then, we may conclude here that

$$\mathrm{SDR}[(\mathbf{Inst.})^{\mathsf{estim}}; (\mathbf{Inst.})^{\mathsf{GT}}] \tag{4.9}$$

$$= \mathrm{NSDR}[(\mathbf{Vocal})^{\mathsf{estim}}; (\mathrm{Mixture}), (\mathbf{Vocal})^{\mathsf{GT}}] \tag{4.10}$$

$$\approx 4\ [\mathrm{dB}]. \tag{4.11}$$

Qualitatively, the performance on the songs of female singers is totally better than those of male singers, though the sound quality of resultant karaoke signals depended on the individual songs.

Sliding HPSS

We may execute HPSS online by the procedure shown in Figure 4.1 and Algorithm 4.2,4.3, with the latency approximately Rs, where s is the frame shift of STFT and I is the size of “sliding block,” which is the randomly-accessible queue, that accepts and emits spectral fragments [112]. The procedure is basically similar to RTISI-LA, as a technical component of key conversion, which is described in Appendix C.

4.4 Implementation of the System

4.4.1 Overview of the Software

In this section we describe the implementation of the system. We implemented the system to receive the input from the line input, instead of streaming input (but essentially similar), and output to the line output. Euterpe consists of the cascade connection of five processing

Technical Components of the System

Audio Input Input audio signals from streaming audio input (or file).

Two-stage HPSS Similar to Chapter 2.

- HPSS 1: Original signal $\xrightarrow{\mathsf{HPSS}(\text{Short})} h'(t), p(t)$
- HPSS 2: $h'(t) \xrightarrow{\mathsf{HPSS}(\text{Long})} h(t), v(t)$

Synchronization and Add Synchronize the output of two-stage HPSS $h(t), v(t), p(t)$, and add them. i.e., $x(t) = \alpha_h h(t) + \alpha_v v(t) + \alpha_p p(t)$, where $\alpha_h, \alpha_v, \alpha_p$ are certain coefficients, controlled by the users through the UI.

Transposition Apply key transposition and wave synthesis. (See Appendix C)

Audio Output Output the signal to the audio device. The real-time audio input/output was achieved by PortAudio [5].

Figure 4.4:: Five blocks in Euterpe System

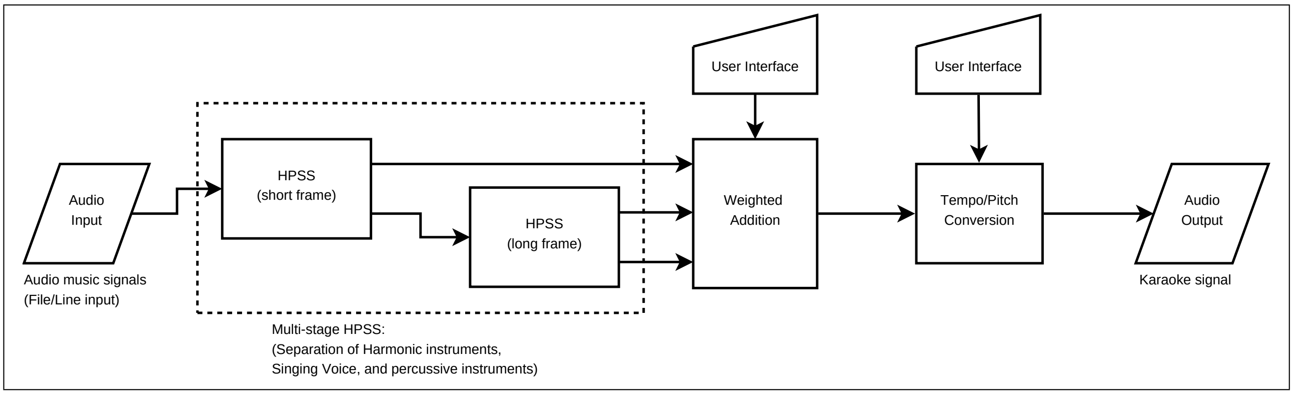


Figure 4.5:: Concept of the whole procedure of Euterpe. The system first separates the audio input into three components by two-stage HPSS into harmonic, singing, and percussive components. The process is followed by the synchronization and weighted summation, because we should add harmonic and percussive components to obtain the “accompaniment.” The weighting constants can be controlled by GUI. Then the system converts the key, and outputs the karaoke signal to the audio device.

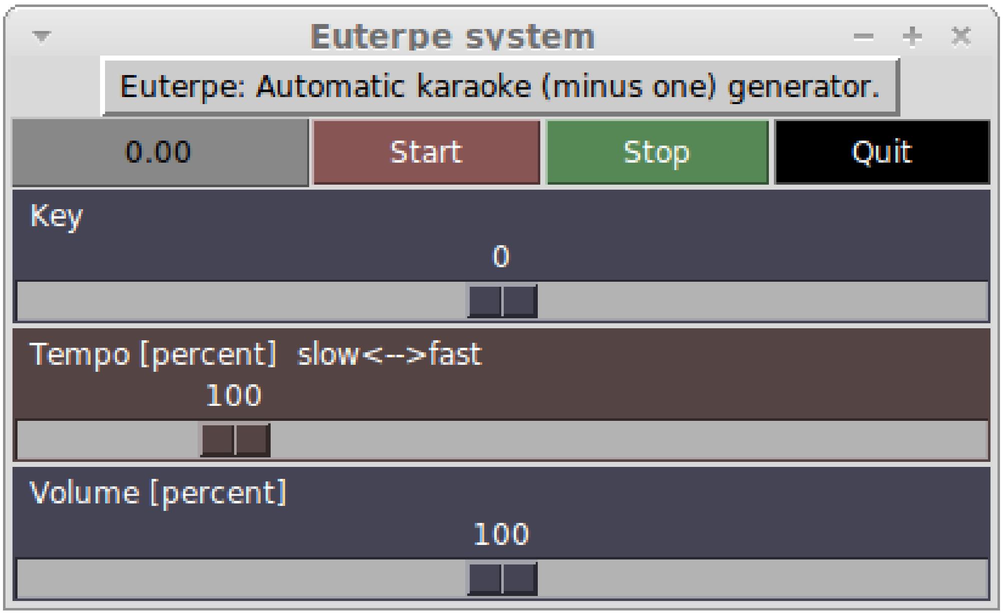

Euterpe system
Euterpe: Automatic karaoke (minus one) generator.
0.00
Start
Stop
Quit
Key
0
Tempo [percent] slow<-->fast
100
Volume [percent]
100

Figure 4.6:: User interface of Euterpe

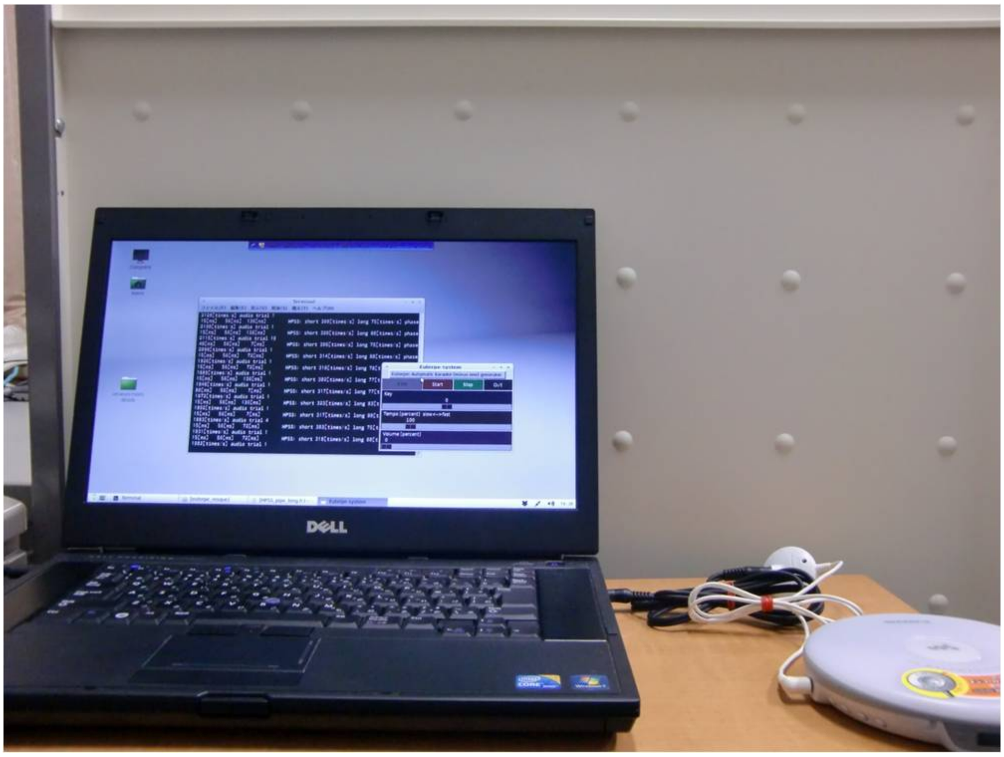


Figure 4.7:: Photograph of experiment system. There are a laptop PC, in which Euterpe system works, and CD player which is connected to the line input of the PC, from which an audio signal is input into the Euterpe system.

blocks, shown in Figure 4.4. The total architecture is based on pipeline model as shown in Figure 4.5.

Figure 4.6 shows the user interface of the system. Euterpe has three buttons and some sliders. Three buttons indicates “start,” “stop,” and “quit,” which are ordinary buttons for music players. It also has some sliders which are also common in music players. By dragging the sliders the users can change the parameters, namely the parameters of HPSS and the key, as well as $\alpha_h, \alpha_p, \alpha_v$ in Figure 4.4.

4.4.2 Remarks

Pipeline-based Implementation

Computation of HPSS and key transposition are not quite costly. In fact, the processings are much faster than the real-time. However, it is still costly operations, when it comes to the critical real-time processing. Then, we designed the architecture by a pipeline model based on multi-thread programming [107, §2.2.3], because of two reasons below.

A reason is the flexibility of buffering between two components. That is, it buffers the difference of the units of I/O. For example, HPSS(short) reads/writes 256 samples at a time, while HPSS(long) reads/writes 1024 samples at a time. Because of the different sized blocking I/O unit, HPSS(long) should wait for HPSS(short) is executed 4 times. Implementation of this kind of processing becomes simpler if it is based on pipeline model.

Another reason is the load dispersion. Because of the computation cost of HPSS and wave synthesis, the processing may not be finished in the callback function of PortAudio [5] API. In callback function, instead of executing these costly calculations, just copying the already-processed data to the output buffer is rather preferable. Pipeline model complies with this requirement.

Sliding Block Analysis and Flexible Iteration

In Algorithm 4.3, line 9–11, we may expect that there should be some chances to update the spectrogram more than R_i times, if we have a computer of higher performance. In this dissertation, in order to increase the number of iteration R_i as large as possible, we considered to make R_i more flexible, instead of setting R_i by a constant.

Table 4.8: Number of iteration of each technical component in 1 [s]. The values are basically the approximate maximal values, because the number of iteration of each component, especially waveform synthesis, often varied depending on the load of other processes, etc.

Machine	Short HPSS	Long HPSS	Pitch Transposition
DELL PRECISION M4500	320 [Times/s]	80 [Times/s]	7000 [Times/s]
Lenovo ThinkPad T400s	320 [Times/s]	80 [Times/s]	4000 [Times/s]
ASUS EeePC	300 [Times/s]	80 [Times/s]	500 [Times/s]
Minimal (= inverse of frame shift)	63 [Times/s]	16 [Times/s]	63 [Times/s]
Frame shift	16 [ms]	64 [ms]	16 [ms]

This is simply achieved to make each agent (i.e., each HPSS, etc) to send inquiries to the connected buffers how long data are still stored in the buffer, then each agent make decision whether it updates one more time or emits the head of the queue to the buffer.

4.4.3 Throughput

We implemented the Euterpe system on the basis of the software architecture above, and verified that it works on following computers, though we sometimes observed some "underrun" and some other errors of ALSA (Advanced Linux Sound Architecture).

Lenovo Think Pad an ordinary laptop PC. OS: Ubuntu.

DELL PRECISION a laptop workstation. OS: Mint Linux.

ASUS EeePC an ordinary netbook. OS: Xubuntu.

Table 4.8 shows the approximate throughput of the system; it displays the approximate number of iterations of each technical component per a second. "Minimal" indicates the least number of iteration each updating formulae must be executed to keep up with the real-time. The table shows that the system works in real-time even on a net book (ASUS EeePC), executing HPSS and wave synthesis routines sufficiently may times.

4.5 Summary of Chapter 4

We developed an automatic karaoke generating system “Euterpe,” which aimed at generating a karaoke signal automatically from a wider range of music signals, including monaural signals, live recordings, compositions which are not based on the customs of the professional musicians.

This system has following two functions. Firstly, it suppresses singing voice from mixed music audio signals. The performance of the singing voice suppression is basically the same as the performance of singing voice enhancement which is shown in Chapter 2 in terms of SDR. Another function of Euterpe is the key modification without changing the timbre of the accompaniment that shall be described in Appendix C.

We also described the architecture of the implementation of the whole systems, which is based on the pipeline model. The advantages of pipeline model include flexibility of blocking I/O and load dispersion. Because of these architecture, we achieved the real-time karaoke generating system, which is especially important when we consider a client-side streaming-based karaoke generating system.

There are still rooms to improve the system, in terms of the sound quality, as well as the “faithfullness” to the ordinary karaoke systems; adding some other functions such as showing lyrics, etc.

Chapter 5

Distribution of Characteristic Fluctuation Time-scale of Signals

5.1 Introduction: Signal Decomposition by “Degree” of Fluctuation

5.1.1 Extension of the Concept of Two-stage HPSS

In Chapter 2 we have verified the effectiveness of applying HPSS twice on differently-resolved spectrograms to extract singing voice from music; we considered to decompose a signal into three classes shown in Figure 5.1. The idea is based on a fact that a sound with fluctuation appears on the two spectrograms differently, depending on the time-frequency resolution of spectrogram, or in other words, the set of the time-frequency bases of the spectrogram; both are parametrized by the frame length L.

Let us assume a hypothesis that a vocal component has a characteristic time-scale of fluctuation L^*, and that we may rewrite the discussion in Chapter 2 that the singing voice was classified into H and P depending on the relation between L^* and the frame length L. That is, if the frame length is much shorter than the characteristic time-scale of singing voice L^*, i.e., $L \ll L^*$, then the fluctuation of scale L^* can be ignored in the scale of L, and it is classified into H component, while if $L \gg L^*$ then the component is classified into P component because it requires many time-frequency bases at a time to represent the

Notes on Chapter 5: The contents of this chapter is the revisions of the authors’ domestic conference papers, [311], [312], [313], [314]. The paper [314] received the Student Presentation Award of Acoustical Society of Japan.

fluctuation shorter than L.

Under the hypothesis, we may naturally expect that the time-scale L^* may be rather precisely specified by trying more and more frame lengths than we used in Chapter 2. That is, instead of considering different two frame lengths L_1 and L_2, we attempt to consider more number of spectrograms of different resolution, $L'_1 < L'_2 < \cdots < L'_m$, in this chapter. Applying HPSS on these spectrograms, a signal is separated in many ways, from which we may naturally construct a "spectral" representation of signal, as Figure 5.2, from which we may find which L'_k is the "boundary" L^b between two classes of frame lengths $A = \{L|L < L^b\}$ and $B = \{L|L > L^b\}$, such that if $L \in A$ then the signal is classified into "harmonic," while if $L \in B$ then the signal is classified into "percussive," by HPSS on $\mathsf{STFT}(L)$; the "boundary" L^b may be interpreted as the "characteristic time-scale" L^* of the signal.

In this chapter, we show a signal decomposition technique based on the idea above, and define a feature vector that may represent the "distribution of the characteristic time-scale" of a signal. We then show some examples of feature vectors of speech and music signals, which have quite distinctive distribution each other.

5.1.2 Related Work

As discussed in section 2.2.2 in pp. 20, similar concepts which capture the fluctuation of signals, namely speech signals, have been proposed, such as modulation spectrum, RASTA [59]; these techniques have been applied in many applications including [58], [73], [74], [80], [91], [141], [147], etc. These are based on the Fourier analysis of the envelope of the signal. That is, it is basically based on the spectral analysis of the amplitude modulation.

In this thesis, we do not intend to claim that the performance of the proposal method outperforms the existing techniques above. Instead, our aim in this chapter is to show another possible approach to the fluctuation of signals, on the basis of the concepts discussed in Chapter 2.

Three Typical Components:

$\uparrow$ **Sinusoidal components**

$\boldsymbol{\mathcal{H}}$ Sinusoidal, flatly-played instruments, (e.g., piano, guitar),

$\boldsymbol{\mathcal{V}}$ Fluctuated quasi-stationary component (e.g., singing voice),

$\boldsymbol{\mathcal{P}}$ Transient, non-stationary instruments, (e.g., percussion).

$\downarrow$ **Instantaneous non-stationary components**

Figure 5.1:: Three typical components in terms of the depth of fluctuation.

Classification into More Than Three Components:

$\uparrow$ **Sinusoidal components**

$\boldsymbol{\mathcal{H}}$ Sinusoidal, flatly-played instruments, (e.g., piano, guitar),

$\vdots$ (components between $\boldsymbol{\mathcal{H}}$ and $\boldsymbol{\mathcal{V}}$)

$\boldsymbol{\mathcal{V}}$ Fluctuated quasi-stationary component (e.g., singing voice),

$\vdots$ (components between $\boldsymbol{\mathcal{V}}$ and $\boldsymbol{\mathcal{P}}$)

$\boldsymbol{\mathcal{P}}$ Transient, non-stationary instruments, (e.g., percussion).

$\downarrow$ **Instantaneous non-stationary components**

Figure 5.2:: Signal decomposition into more than three typical components.

5.2 Extension of Two-stage HPSS into Concurrent Multiple HPSS

5.2.1 Two concurrent HPSS

Let us recall Chapter 2. We have considered to analyze a fluctuating signal on spectrograms with different frame lengths, $L_1 = 30$ [ms], $L_2 = 500$ [ms]. On the former spectrogram, i.e., $L_1 = 30$ [ms], the fluctuating signal occupies many time frames, while it does not occupy many frequency bins. On the other hand, on the latter spectrogram, i.e., $L_1 = 500$ [ms] each component occupies many frequency bins, while it does not occupy many time frames. Thus, the directions of continuity of the spectrogram are different in two spectrograms, and they are separated quite differently by HPSS.

We may write the processings above as follows,

$$s(t) \xrightarrow{\mathsf{HPSS}(L_1)} h_1(t), p_1(t) \tag{5.1}$$

$$s(t) \xrightarrow{\mathsf{HPSS}(L_2)} h_2(t), p_2(t). \tag{5.2}$$

where singing voice is mostly included in $h_1(t)$ and $p_2(t)$. The reason why singing voice separated into these components, in our hypothesis, is that the singing voice has characteristic time-scale between L_1 and L_2. In fact, singing voice has fluctuation of interval around 100 [ms], which is greater than L_1 and smaller than L_2.

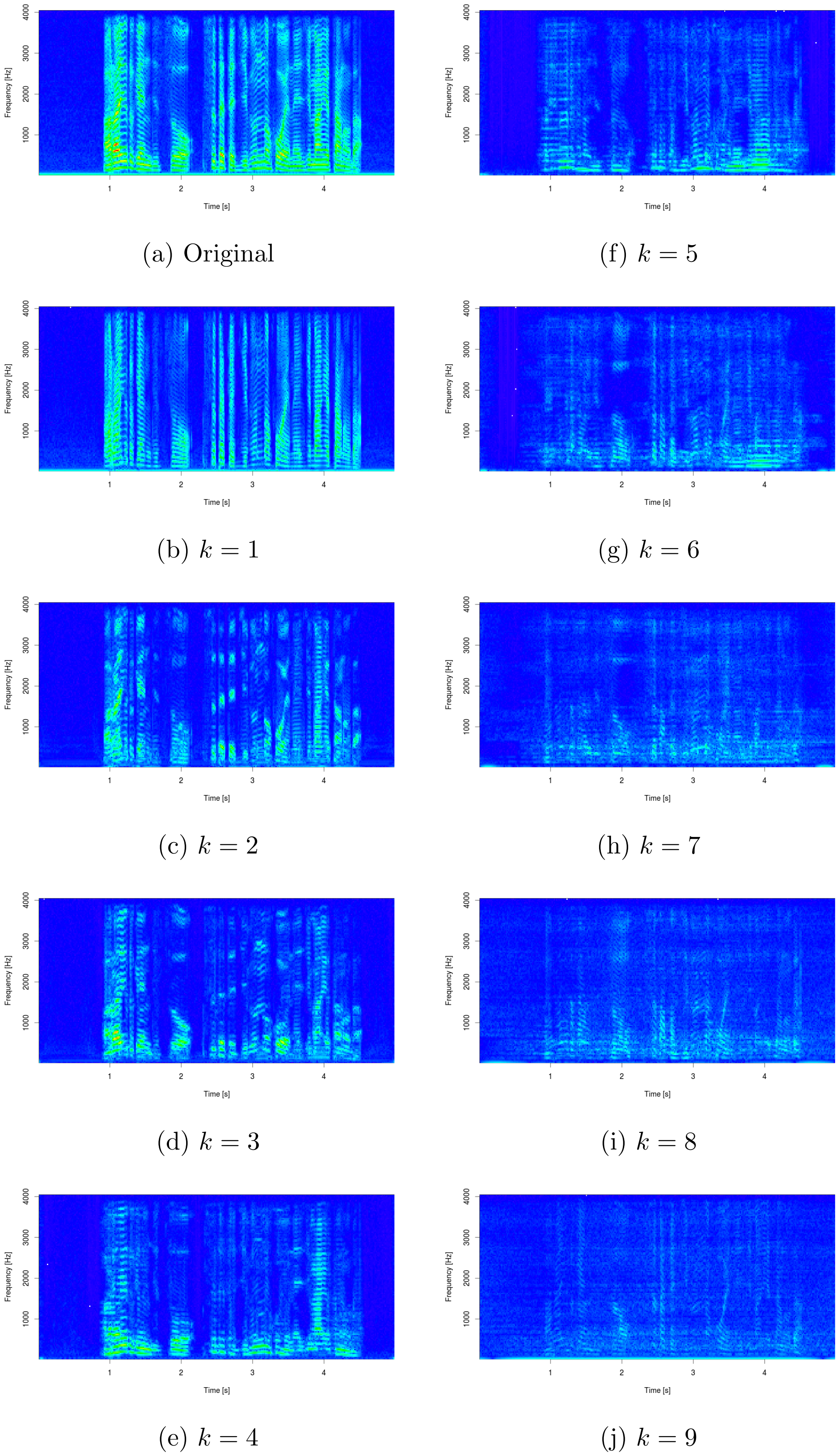
5.2.2 Concurrent Multiple HPSS

On the basis of the idea above, it would be natural to consider more number of frame lengths, i.e., time-frequency resolution, and consider signal decomposition using them. Let $\boldsymbol{S}_k (k = 1, 2, \ldots, n)$ be spectrograms of an input signal $s(t)$. The spectrogram $\boldsymbol{S}_k$ is defined by the STFT, using an analyzing frame length L_k, as follows,

$$\boldsymbol{S}_k = \mathsf{STFT}(L_k)\left[s(t)\right], \tag{5.3}$$

where $L_1 < L_2 < \cdots < L_n$.

Similarly to the previous subsection, HPSS is applied to each spectrogram individually.



(a) Original

(f) $k = 5$

(b) $k = 1$

(g) $k = 6$

(c) $k = 2$

(h) $k = 7$

(d) $k = 3$

(i) $k = 8$

(e) $k = 4$

(j) $k = 9$

Figure 5.3:: Spectrograms of CMHPSS components. (a) Original speech signal $s(t)$ (a male speech signal in JNAS database) (b)–(j) Spectrograms of CMHPSS components $\mathsf{CMHPSS}(k)[s(t)]$, where $L_k = 64 \times 2^k/8000$ [s], $k = 1, \ldots, 8$

Therefore, we may consider the following decompositions. That is, applying HPSS to $\boldsymbol{S}_k$,

$$s(t) \xrightarrow{\mathsf{HPSS}(L_1)} h_1(t), p_1(t) \tag{5.4}$$

$$s(t) \xrightarrow{\mathsf{HPSS}(L_2)} h_2(t), p_2(t) \tag{5.5}$$

$$\vdots$$

$$s(t) \xrightarrow{\mathsf{HPSS}(L_n)} h_n(t), p_n(t) \tag{5.6}$$

and n "percussive" components $p_k(t), (1 \leq k \leq n)$ are obtained.

In our hypothesis, $p_k(t)$ contains the components whose characteristic time-scale is shorter than L_k. On the basis of the assumption, using thus obtained n "percussive" components, we may roughly extract components, whose characteristic time-scale of fluctuation would be between L_{k-1} and L_k, by

$$\mathsf{CMHPSS}(k)[s(t)] := \begin{cases} p_1(t) & k = 1 \\ p_k(t) - p_{k-1}(t) & 2 \leq k \leq n \\ s(t) - p_n(t) & k = n+1. \end{cases} \tag{5.7}$$

Note $s(t) = \sum_k \mathsf{CMHPSS}(k)[s(t)]$. We call this decomposition as Concurrent Multiple HPSS (CMHPSS).

Figure 5.3 shows an example of the decomposition above, in a case of a speech signal. In the figure, it is observed that most components in voice are contained in the components $1 \leq k \leq 4$ where $L_k = 64 \times 2^k/8000$ [s], and we may discuss that the speech signal is composed of components whose characteristic time-scale is around $128/8000 (= 0.016)$ [s] to $1024/8000 (= 0.128)$ [s].

5.3 Characteristic Fluctuation Time-scale Distribution

5.3.1 Power Distribution of CMHPSS

In our assumption, each component $\mathsf{CMHPSS}(k)[s(t)]$ roughly correspond to a component, whose characteristic time-scale of fluctuation is roughly L_{k-1} to L_k. Therefore, using these components we can construct a feature vector to measure the distribution of fluctuation of the signal. Let us define it as follows.

$$\boldsymbol{S} = (S_k)_{1 \le k \le n} = (S_1, S_2, \dots, S_n), \tag{5.8}$$

$$\text{where} \quad S_k = \left\| \mathsf{CMHPSS}(k)[s(t)] \right\|^2 \tag{5.9}$$

$$= \sum_t \left\{ \mathsf{CMHPSS}(k)[s(t)] \right\}^2 \tag{5.10}$$

It is reasonable to consider that the power of each component of the vector $\boldsymbol{S}$, i.e. S_k, represents the strength of each component, whose characteristic time-scale is L_k. Let us simply call the feature as Characteristic Fluctuation Time-scale (CFTS) in this chapter.

5.3.2 CFTS Distribution of Speech and Music Signals

We show some examples of CFTS. The conditions are as follows; the sampling rate of all the data was 8 kHz; all the signals were monaural; the length of all the signals were 10 [s]. (When the length of a signal was less than 10 [s], we concatenated the signal several times.) The condition of HPSS is as follows; STFT frame lengths were $L_k = 64 \times 2^k, (1 \le k \le 8)$, i.e., $16, \cdots, 2048$ [ms]; frame shift was half the length of frame lengths. The parameter of HPSS was $\sigma_{\mathrm{H}} = \sigma_{\mathrm{P}} = 0.3$, and the size of sliding block (i.e., the number of iteration) was 30. These conditions are identical to the original HPSS except the frame length.

CFTS distribution of Speech Signals

Figure 5.4, 5.5 show the CFTS distribution of the speech signals of 10 male speakers and 10 female speakers, respectively. The speech signals were excerpted from JNAS [70] database.

All of these 20 speech segments has characteristic CFTS distribution; they have commonly a strong component of around the time-scale 0.1 [s] $(k = 3, 4)$. Another characteristic is that the components around the time-scale around 1 [s] are weaker $(k = 5, 6, 7)$.

CFTS distribution of Music Signals

Figure 5.6 shows CFTS distributions of piano music, and Figure 5.7 shows CFTS distributions of the 10 orchestral music, both are excepted from RWC database [46], RWC-MDB-C-2001 No. 26 – No. 35, and RWC-MDB-C-2001 No. 1 – No.10, respectively.

It is observed that they have typically strong component in around the time-scale of around 1[s] $(k = 5, 6, 7)$. It is also observed that the CFTS distribution of all of these 20 signals are

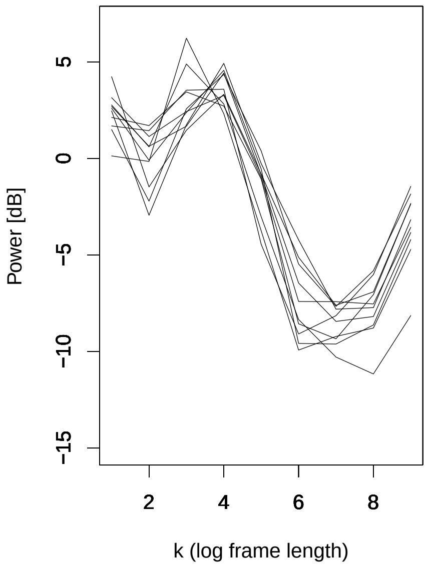


Figure 5.4:: CFTS distribution of 10 male speech

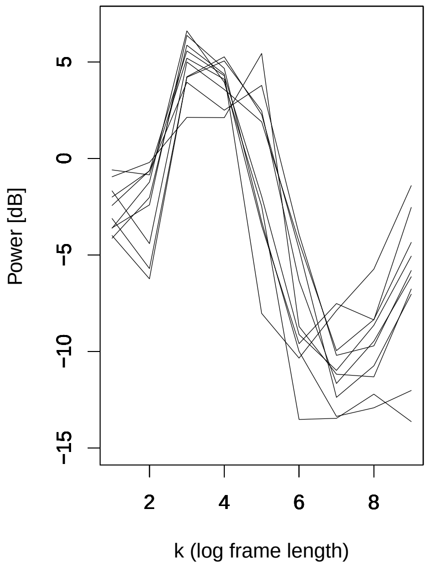


Figure 5.5:: CFTS distribution of 10 female speech

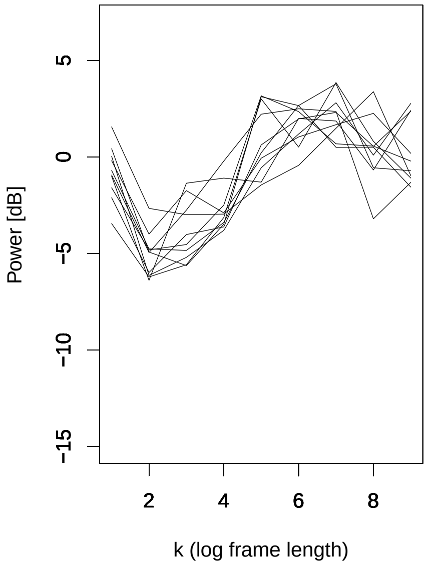


Figure 5.6:: CFTS distribution of 10 piano music signals

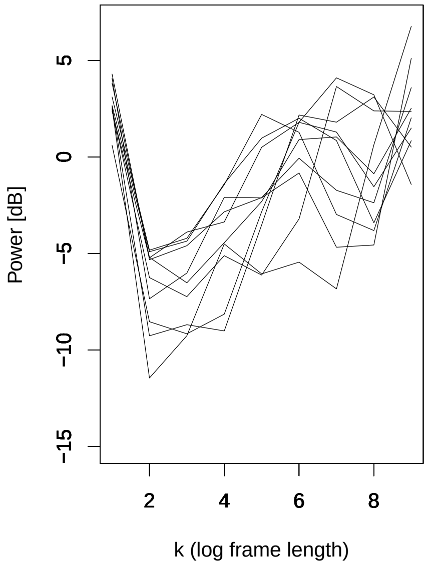
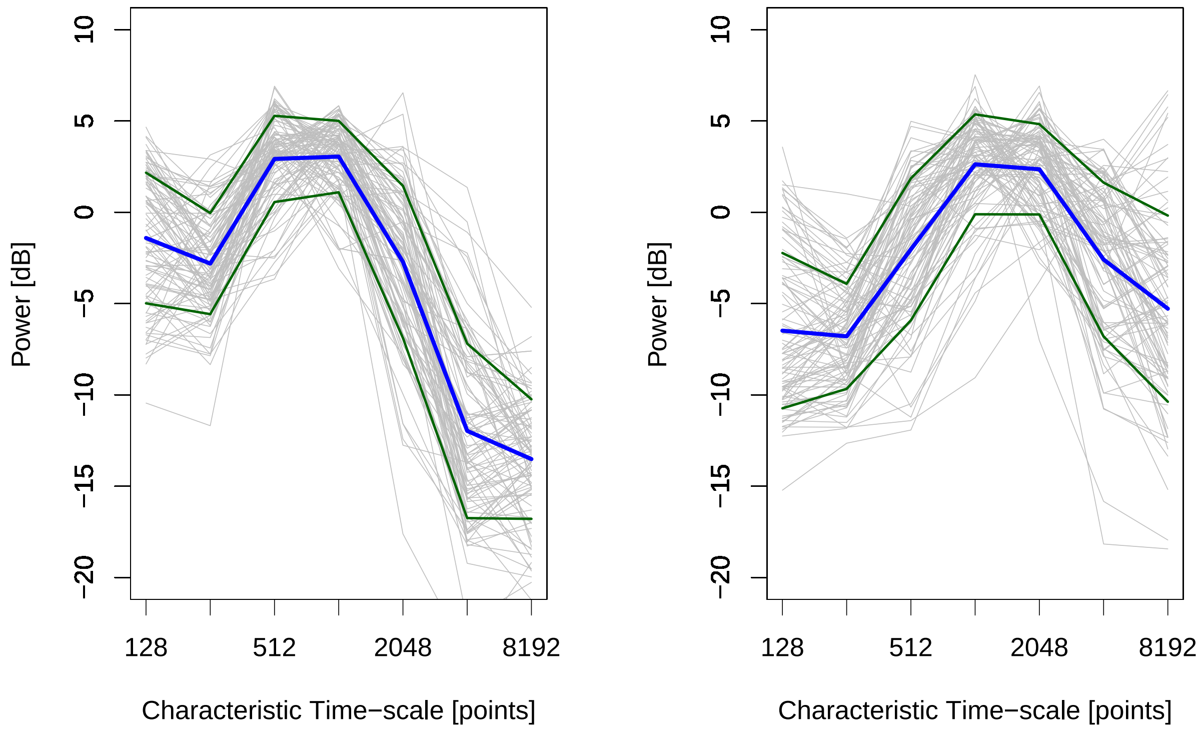


Figure 5.7:: CFTS distribution of 10 orchestral music signals



(a) (b)

Figure 5.8:: CFTS distribution of speech and singing voice. (a) speech signals (b) singing voice signals. Blue thick lines indicate the average of 100 CFTS distributions, and the green lines indicate the standard deviation of them.

distinctively different from those of the speech signals, which suggests that it can be used as a feature to discriminate them.

5.3.3 CFTS Distribution of Speech and Singing Voice

Figure 5.8 shows the examples of the CFTS feature vector, of 100 speech signals and 100 singing voice signals. Each CFTS feature vector was obtained under the following condition: the sampling rate was $f_\mathrm{s} = 8000$ Hz, and the frame lengths were $L_k = 64 \times 2^k$ points ($64 \times 2^k / f_\mathrm{s}$ [s]), where $k = 1, 2, \cdots, 6$.

Comparing the feature vectors of speech and singing voice, it is observed that CFTS feature vector of speech and singing voice have distinctly different shapes, despite the variety of the speakers and singers, as well as the variety of the texts they speak and sing. This fact also suggests that thus obtained CFTS may be an effective feature to discriminate speech and singing voice.

5.4 Example: Speech/Singing Discrimination

5.4.1 Discrimination of Singing voice and Speech

As an application, we consider the discrimination task of speech (talk) and singing voice, which would be a basic technical component of an audio search engine.

Although there have been some studies on music/speech discrimination, e.g. [29], [139], the discrimination of singing and speech may not be similarly solved because the difference between singing voice and speech is not as distinctive as that of between instruments and speech. In other words, we may not exploit the nature of instruments in this case, which was available in the case of music/speech discrimination.

According to the experiments by Ohishi et al. [109], [110], humans have ability to recognize a two-second segment singing or speaking, in an accuracy of 99.7%. Ohishi et al. [109], [110] then proposed a technique that discriminate two-second segments of singing voice and speech in the accuracy of 87%, by exploiting MFCC and F_0 information.

In this chapter, we show that CFTS can be used another feature for this purpose. We show that there is distinctive difference between CFTS distributions of both speech and singing voice. Experimental results show that the performance of the discrimination using CFTS as a feature outperformed that of using only MFCC.

5.4.2 Experimental Condition

We conducted an experiment on speech/singing voice discrimination to verify that CFTS distribution is an effective feature of audio signal discrimination. In the experiment, we first computed CFTS distributions of the given input speech signals and singing voice signals. We then applied a classification algorithm using this distribution as a feature vector.

Dataset

Singing voice signals were excerpted from MIR-1K database [62][⋆1]. Each data were divided into two-second segments. Note that we removed a number of segments from the datasets,

[⋆1] http://sites.google.com/site/unvoicedsoundseparation/mir-1k

Table 5.9: The accuracy of speech/singing discrimination. Chance level is 50.0%.

Feature	Accuracy
MFCC (including log energy, Δ, and $\Delta\Delta$) (39-dim)	85.5%
CFTS distribution (7-dim)	92.1%
CFTS distribution (7-dim) + MFCC (39-dim)	**93.8%**

because there were many data with long silence, which is not suitable for the experiment. We removed such data from the dataset automatically, and thus we have derived 3461 singing voice segments of length 2 [s], which is equivalent to 1h55m22s.

Speech data were extracted from Japanese Newspaper Article Sentences database (JNAS). Each signal was also divided into two-second segments, and inadequate data were automatically removed by the same manner described above. Thus more than 80,000 speech segments of length 2 [s] were obtained, and we extracted the same number of segments randomly from them.

Note that the sampling rate of all the data above was 8000 Hz. We calculated the CFTS distribution of each signal similarly to the way we described above, i.e., $L_k = 64 \times 2^k, (k = 1, \ldots, 6)$.

Referential Features for Comparison

As a referential feature, we used 12-dim MFCC + log energy + Δ + $\Delta\Delta$ (39-dim in sum), which would be the most basic and commonly-used audio signal feature. We calculated MFCC using HTK[⋆2]. We also considered to use both of CFTS (7-dim) and MFCC (39-dim).

Classifier and Evaluation Criteria

We used AdaBoost [38] as a classifier. The weak classifier used in AdaBoost was linear discriminator. Under this condition, we executed the 10-fold cross validation, and evaluated the accuracy rate of speech/singing discrimination.

5.4.3 Results

Table 5.9 shows the accuracy of speech/singing discrimination. When we used CFTS distribution as the feature, the discrimination accuracy was 92.1%. The digit is higher than 85.5%, which is the accuracy of MFCC-based discrimination.

[⋆2] [Online (Visited Jan. 2014)] `http://htk.eng.cam.ac.uk`

The joint use of both features was better than both cases. The accuracy ratio was 93.8%, which is much higher than the accuracy when we used only MFCC-based features. This result indicates effectiveness of CFTS distribution as a feature for speech/singing voice discrimination.

5.5 Summary of Chapter 5

In this chapter, by extending the idea in Chapter 2, we have discussed a signal decomposition technique based on its characteristic time-scale of fluctuation. The algorithm consists of HPSS on multiple spectrograms which have different time-frequency resolution. By running HPSS concurrently on these spectrograms, we obtained many "harmonic" and "percussive" components, and by subtracting them we obtained the decomposed signals. We called the whole procedure CMHPSS.

We then defined CFTS features, which is an array of each power of CMHPSS components. CFTS distributions of speech signals showed rather similar distributions, while they were clearly distinctive from music signals.

In this chapter we applied CFTS to a speech/singing discrimination problem, which would be a fundamental technique for audio search engine. The method is based on the fact that the singing voice and speech have quite different CFTS distributions. We used them as a feature and AdaBoost as a classifier to discriminate speech and singing voice, and showed that the accuracy of the discrimination outperformed the case in which we used MFCC as a classifier, a standard audio feature. Furthermore we showed that the joint use of both MFCC and CFTS features improved the performance.

Chapter 6

Discussions on HPSS

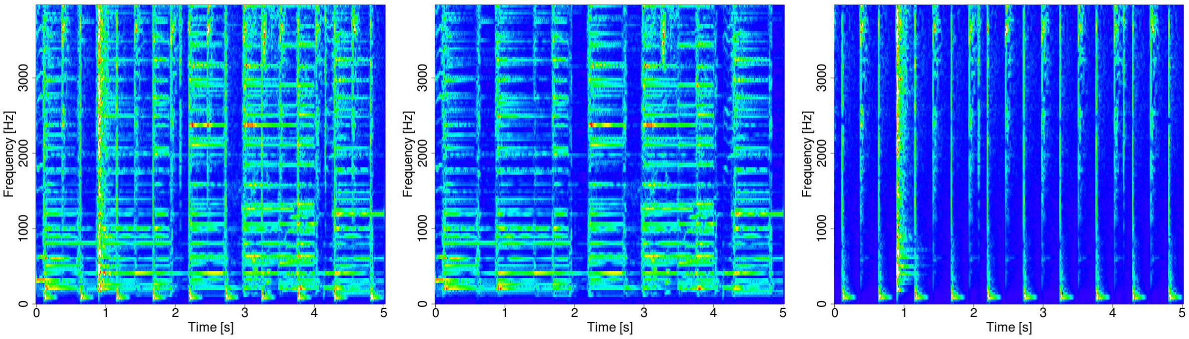
6.1 Introduction

In this section, we discuss the details of Harmonic/Percussive Sound Separation (HPSS) which was shown in a nutshell in Chapter 2. In addition to describing HPSS in detail, we also consider the improvement of the formulation of HPSS, because the technique forms the basis of our singing voice enhancement technique and other relevant techniques, as we have seen in this dissertation. We may naturally expect that improving HPSS may result in better performance of the techniques described in the previous chapters. Specifically, the algorithm of HPSS has some potential to be improved such as following:

- Accelerate the performance taking long-term relation into account.
- Replace the reconstructivity cost function by strict constraints to simplify the formulation.
- Some desirable properties for optimization problems are not guaranteed in the original HPSS, such as global optimality and uniqueness of the solution.

Notes on Chapter 6: This chapter consists of some materials of an authors' paper [301] and an authors' domestic conference papers [306]. Some parts of this chapter, namely the concept of anisotropic smoothness (Section 6.2.1) and one of the algorithms (Section 6.4.3), are basically based on Ono's conference paper [112], though the author have rewritten the text.

The concept "anisotropic smoothness" in this chapter was first shown in conference papers [101], [100], [113], [112], in which I was not included in the authors. The author's notable contributions in this chapter are as follows: (1) Reformulated the algorithm, and obtained "Method 1-A" and "1-B." (2) Considered general N' and K', which were $N' = K' = 1$ in [112] etc.,



(a) (b) (c)

Figure 6.1:: Spectrograms of (a) a mixed music signal $\boldsymbol{Y}$ (extracted from a song "dreams"), (b) a harmonic component $\boldsymbol{H}$, in which it is observed that it is rather continuous in time than in frequency, and (c) a percussive component $\boldsymbol{P}$, which is continuous in frequency. (This figures are identical to Figure 2.5 in page 23.)

This chapter considers issues above, i.e., describing HPSS in detail, and considers other formulations of HPSS. Note, many of the contents in this chapter are almost independent of the discussion on singing voice.

Related Work Since the purpose of this chapter is to consider HPSS itself, we do not review relevant techniques here, but we summarized the related work on harmonic/percussive separation in Appendix B.

6.2 Concept of HPSS

6.2.1 Anisotropic Smoothness of Spectrogram

An amplitude spectrogram $\boldsymbol{Y} = |\hat{\boldsymbol{Y}}| \in \mathbb{R}^{N \times K}$ of a typical music signal $y(t)$ is shown in Figure 6.1 (a). We may see that the spectrogram has a check pattern, composed of crossing horizontal lines and vertical lines. The reason why a musical spectrogram has such a pattern is that the music signals are typically composed of two typical classes of instruments, i.e., the harmonic and the percussive.

It is likely that the horizontal (temporally-continuous) components in Figure 6.1 (a) are attributable to some instruments such as a guitar, a piano, etc., noting that the amplitude spectrograms of these sounds are likely to be temporally smooth as shown in Figure 6.1 (b), because of their quasi-stationarity. To be specific, let $\boldsymbol{H} \in \mathbb{R}^{N \times K}$ be an amplitude spectrogram of harmonic sounds, and we may assume that the value of the spectrogram at a time-frequency bin (n, k), i.e. $H_{n,k}$, should be nearly equal to those of the temporally adjacent bins $(n \pm 1, k)$. That is,

$$H_{n,k} \approx H_{n\pm1,k}. \tag{6.1}$$

Note, not entirely identical but essentially similar properties are supposed on harmonic instruments in [21], [28], [33], [157], [158], [159], [160], etc. Similarly, it is likely that the vertical (continuous in frequency) components are attributed to percussive instruments, noting that the spectrogram of percussive instruments are likely to be smooth in frequency as shown in Figure 6.1 (c), because of their impulse-like nature.

$$P_{n,k} \approx P_{n,k\pm1}. \tag{6.2}$$

In summary, harmonic and percussive components are continuous anisotropically. On the basis of the discussion above, we may expect that applying an algorithm that separates a crossing check pattern into horizontal and vertical components on spectrogram can separate harmonic and percussive components of music signals. This is the fundamental concept of HPSS.

Long-range Anisotropic Smoothness

As described above, the conventional HPSS is formulated on the basis of the relation to the neighboring bins $(n-1, k)$, $(n+1, k)$, $(n, k-1)$, $(n, k+1)$ in order to evaluate the smoothness of the spectrogram $\boldsymbol{H}$ and $\boldsymbol{P}$, respectively. However, these assumptions may not be necessarily effective enough to evaluate the smoothness of spectrogram, because the neighboring bins $(n-1, k)$, $(n+1, k)$, $(n, k-1)$, $(n, k+1)$ are not sufficiently non-correlated to the bin (n, k), because of the frame overlap of STFT, and the effect of the frequency blur attributable to the window function. In other words, any spectrograms can be "smooth" both in time and in frequency to some extent, if we only considered the just neighboring bins.

On the basis of the motivation above, we may naturally extend HPSS using the long-term relations of spectrograms. Longer-term bins, which are fairly distant from (n, k), would be sufficiently non-correlated to the (n, k) unless the spectrograms are truly "smooth," and hence, these bins are supposed to be more suitable to evaluate the smoothness of spectrograms. That is,

$$H_{n,k} \approx H_{n\pm n',k}, \quad (1 \leq n' \leq N'), \tag{6.3}$$

where N' is the maximal distance we consider neighbour. In the new smoothness function, not only the relation to the neighboring bin $(n \pm 1, k)$ but also the relation to long-term bins such as $(n \pm 2, k), (n \pm 3, k), \cdots, (n \pm N', k)$ are taken into account. The value of N' is supposed to be from 1 to several dozen, from the observations on the spectrogram Figure 6.1 (b) in which it is shown that each sound is typically sustained for $100 - 1000$ [ms], which is equivalent to the several dozens of the bins, if the temporal resolution of STFT is around 10 [ms]. Similarly we may assume $\boldsymbol{P} \in \mathbb{R}^{N \times K}$ should have following property similarly to (2),

$$P_{n,k} \approx P_{n,k \pm k'}, \quad (1 \leq k' \leq K') \tag{6.4}$$

where K' is the maximal distance under consideration.

6.2.2 Criteria on Anisotropic Smoothness

In order to go further from the qualitative discussion above into quantitative discussions, let us define criteria to evaluate how strongly (6.3) and (6.4) are satisfied.

We first define the quantitative criteria on the anisotropic smoothness of spectrogram around each bin (n, k). Although there could be many variants of the way of measuring the "smoothness," we simply defined the criteria as the sum of squared difference between the bins under consideration as follows,

$$S_{\mathsf{time}}(n, k, \boldsymbol{H}^{\gamma}) := \frac{1}{N'} \sum_{n'=1}^{N'} ({H_{n,k}}^{\gamma} - {H_{n-n',k}}^{\gamma})^2, \tag{6.5}$$

$$S_{\mathsf{freq}}(n, k, \boldsymbol{P}^{\gamma}) := \frac{1}{K'} \sum_{k'=1}^{K'} ({P_{n,k}}^{\gamma} - {P_{n,k-k'}}^{\gamma})^2, \tag{6.6}$$

where the superscript γ is an exponential factor to suppress the effects from loud components. Noting that $\gamma \approx 0.6$ roughly approximates human auditory systems in some conditions [140], it would be considerable to set the value around 0.6.

When the condition (6.3) is satisfied, $S_{\mathsf{time}}(n, k, \boldsymbol{H}^{\gamma})$ should take a small value. Similarly, the smoothness of $\boldsymbol{P}^{\gamma}$ in frequency direction around the bin (n, k) can be evaluated by (6.6). In summary,

$$\text{(6.3) is satisfied} \approx S_{\mathsf{time}}(n, k, \boldsymbol{H}^{\gamma}) \text{ is small,} \tag{6.7}$$

$$\text{(6.4) is satisfied} \approx S_{\mathsf{freq}}(n, k, \boldsymbol{P}^{\gamma}) \text{ is small.} \tag{6.8}$$

Using these functions that indicate anisotropic smoothness around a single bin, let us define a "total smoothness" functions, that indicate the smoothness of whole the spectrogram.

Specifically, we defined them by simple summations of the values of $N \times K$ anisotropic smoothness criteria as follows,

$$S_{\mathsf{time}}^{\mathsf{total}}(\boldsymbol{H}^{\gamma}) := \sum_{n} \sum_{k} S_{\mathsf{time}}(n, k, \boldsymbol{H}^{\gamma}) \tag{6.9}$$

$$S_{\mathsf{freq}}^{\mathsf{total}}(\boldsymbol{P}^{\gamma}) := \sum_{n} \sum_{k} S_{\mathsf{freq}}(n, k, \boldsymbol{P}^{\gamma}). \tag{6.10}$$

Let us verify the validity of the criteria (6.9) and (6.10), using real instrumental sounds. The audio files we used for the evaluation were excerpted from RWC-MDB-I-2001 [46] database. Figure 6.2 shows the values of $S_{\mathsf{time}}^{\mathsf{total}}(\boldsymbol{X}^{\gamma})$ defined by (6.9) and $S_{\mathsf{freq}}^{\mathsf{total}}(\boldsymbol{X}^{\gamma})$ defined by (6.10) for each instrument, where $\boldsymbol{X}^{\gamma}$ is a spectrogram such as a piano, a harpsichord, etc. It is shown that harmonic instruments make (6.9) small, (6.10) large, and (6.9) $\ll$ (6.10) is satisfied. To the contrary, percussive instruments make (6.9) relatively large, and (6.10) relatively small. The fact indicates that it is reasonable to use (6.9) and (6.10) as indicators to measure the anisotropic smoothness of the spectrogram $\boldsymbol{X}^{\gamma}$ in time and in frequency, respectively.

6.2.3 Smoothness Function

We finally define an integration of two criteria (6.9) and (6.10) as follows,

$$S(\boldsymbol{H}^{\gamma}, \boldsymbol{P}^{\gamma}; w) := S_{\mathsf{time}}^{\mathsf{total}}(\boldsymbol{H}^{\gamma}) + w S_{\mathsf{freq}}^{\mathsf{total}}(\boldsymbol{P}^{\gamma}), \tag{6.11}$$

where w is a weighting constant. Hereafter, let us call $S(\boldsymbol{H}^{\gamma}, \boldsymbol{P}^{\gamma}, w)$ simply a smoothness function. Note the form of smoothness function S is identical to a part of objective functions of Ono's early studies and the HPSS described in Chapter 2 when $N' = K' = 1$.

Let us consider a thought experiment to make thus defined S as small as possible, under the condition that we are given information whether a bin ${Y_{n,k}}^{\gamma}$ is "harmonic predominant," "percussive predominant," or "silent" for each bin. In this case, it would be reasonable to expect that *classifying harmonic-predominant bins into $\boldsymbol{H}^{\gamma}$ and percussive-predominant bins into $\boldsymbol{P}^{\gamma}$ results in smaller S.*

To return from the digression, what we consider in this thesis is actually the reverse of this. That is, we are expecting that *by minimizing the smoothness function $S(\boldsymbol{H}^{\gamma}, \boldsymbol{P}^{\gamma}; w)$, most of harmonic and percussive instruments may be classified into $\boldsymbol{H}^{\gamma}$ and $\boldsymbol{P}^{\gamma}$, respectively.* In other words, this thesis considers to make computers to separate two components with the

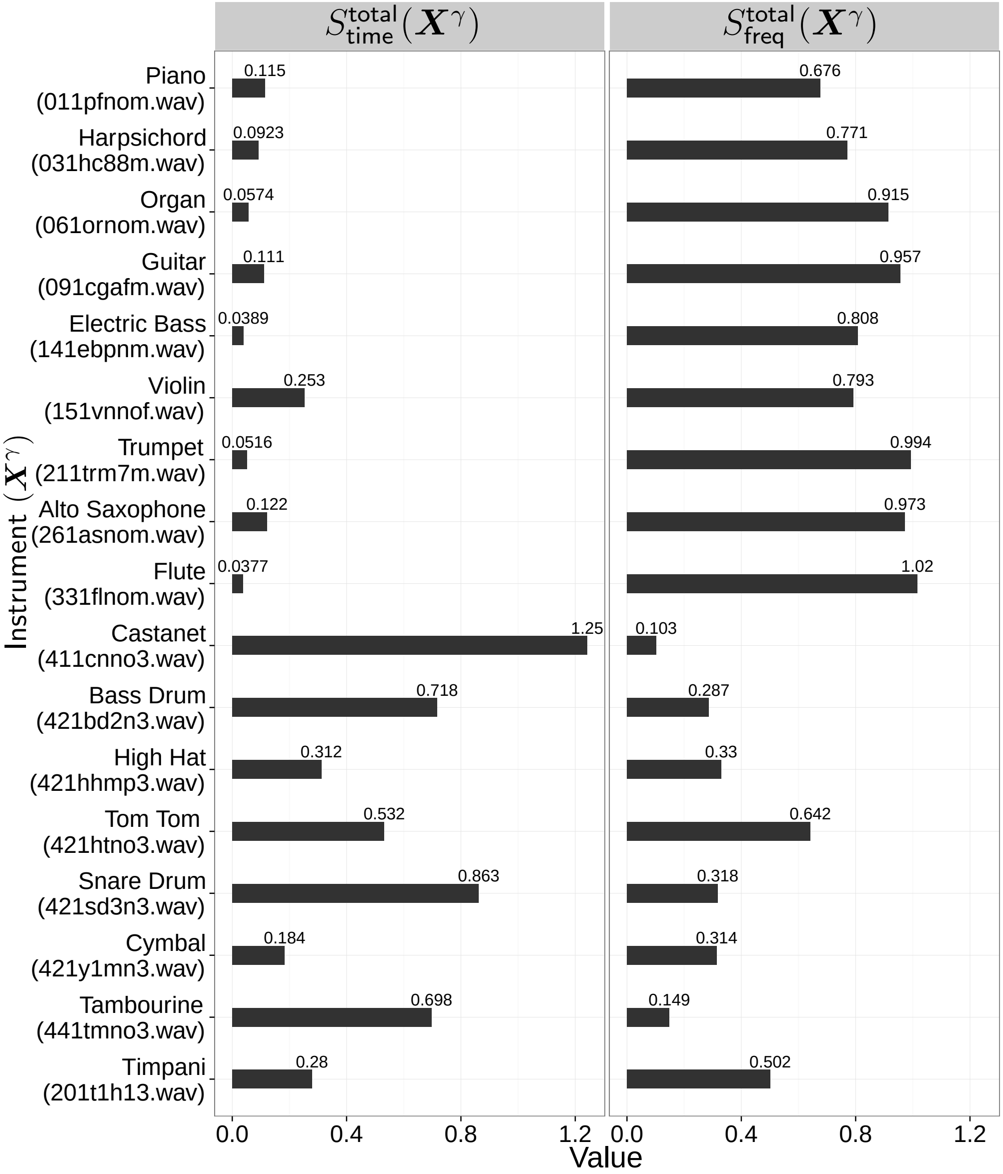


Figure 6.2:: Values of smoothness functions $S^{\text{total}}_{\text{time}}(\boldsymbol{X}^{\gamma})$ and $S^{\text{total}}_{\text{freq}}(\boldsymbol{X}^{\gamma})$. All the values are normalized by the length and the averaged power of each clip. The condition is as follows: $\gamma = 0.5, N' = K' = 5, L = 1024/16000$ [s], frame shift was $L/4$, the window was hanning window, and the sampling rate was 16 kHz. Each instrumental sound was excerpted from RWC instrument sound database [46]. The figure indicates that temporal smoothness function $S^{\text{total}}_{\text{time}}(\boldsymbol{X}^{\gamma})$ defined by (6.9) are quite small for harmonic sounds (piano, harpsichord, ..., flute), while they are relatively large for percussive sounds (castanet, ..., timpani). To the contrary, the values of frequency smoothness function $S^{\text{total}}_{\text{freq}}(\boldsymbol{X}^{\gamma})$ defined by (6.10) are quite large for harmonic sounds, while they are rather small for percussive sounds.

help of the values of $S(\boldsymbol{H}^{\gamma}, \boldsymbol{P}^{\gamma}; w)$, expecting that the minimizer of S may roughly be the harmonic and percussive spectrograms, respectively.

On the basis of the above idea, the source separation problem can be interpreted as an optimization problem like as follows.

Problem (Pre-prototype):

$$\text{minimize} \quad S(\boldsymbol{H}^{\gamma}, \boldsymbol{P}^{\gamma}; w)$$

Note, however, this formulation does not work effectively. Indeed, the simplistic idea of optimizing S just ends up with entirely meaningless results, such as $H_{n,k} = P_{n,k} = -1$, etc. Thus we must avoid this obvious inconvenience by rewriting the problem as follows taking some additional constraint into account.

Problem (Prototype):

$$\begin{aligned} \text{minimize} \quad & S(\boldsymbol{H}^{\gamma}, \boldsymbol{P}^{\gamma}; w) + \text{additional cost} \\ \text{subject to} \quad & \text{some constraints} \end{aligned}$$

One of the most basic costs/constraints is the *non-negativity of each component*, i.e.,

$$\boldsymbol{H}^{\gamma} \geq 0, \boldsymbol{P}^{\gamma} \geq 0, \tag{6.12}$$

because "amplitude" $H_{n,k}$ cannot be negative, which implies the non-negativity of ${H_{n,k}}^{\gamma}$.

Another important cost/constraint is the *reconstructivity*: the sum of separated spectrograms $\boldsymbol{H}$ and $\boldsymbol{P}$ should be almost identical to the original spectrogram $\boldsymbol{Y}$. In the next section, we shall describe the explicit forms of this requirement, as well as the complete problem settings.

6.3 Formulation of HPSS based on the Anisotropic Smoothness

This section describes the explicit formulations of optimization problems that were outlined in the previous section. Specifically, we shall describe three optimization problems.

The difference among these problems is how the requirement on *reconstructivity*, i.e. the cost/constraint on the sum of separated spectrograms, is handled. The following describes the details.

6.3.1 Formulation 1: Considering Reconstructivity as a Constraint

A natural way to lay a constraint on the sum is the restriction of the feasible region. In specific, we may write the constraint as follows,

$$\boldsymbol{H}^{\xi} + \boldsymbol{P}^{\xi} = \boldsymbol{Y}^{\xi}, \tag{6.13}$$

where ξ is an exponential factor, which should satisfy following properties, along with γ:

1. In order to make the condition (6.13) physically meaningful, ξ should be 1, 2, or some values near them. Otherwise another requirement of *exclusiveness* is required.
   (a) When $\xi = 1$, the constraint indicates the additivity of amplitude spectrogram, which implies the wave-domain additivity $h(t) + p(t) = y(t)$, where $h(t)$ and $p(t)$ are harmonic and percussive signals, respectively, under the assumption that the phase spectra are equal, i.e., $\angle\hat{\boldsymbol{H}} = \angle\hat{\boldsymbol{P}} = \angle\hat{\boldsymbol{Y}}$, where $\hat{\boldsymbol{X}} = \mathsf{STFT}(L)\,[x(t)] \in \mathbb{C}^{N\times K}$
   (b) When $\xi = 2$, the constraint means the additivity of power spectrogram.
   (c) General values other than $\xi = 1, 2$ are also acceptable, if exclusiveness condition
   $$(H^{\xi}_{n,k}, P^{\xi}_{n,k}) \approx (Y^{\xi}_{n,k}, 0) \text{ or } (0, Y^{\xi}_{n,k}) \tag{6.14}$$
   is satisfied in many bins, which would be indeed a fair assumption in music spectrogram because it is rather sparse.
2. In order to suppress the effects of outstandingly loud components, the spectrograms should be suppressed by an exponential factor γ. The value should be less than 1, typically around 0.6 which is said to give a fair approximation of human auditory systems.
3. Mathematical convenience requires $\xi = \gamma$ or $\xi = 2\gamma$.

Considering these requirements for γ and ξ, it may be reasonable setting $\xi = 2\gamma$, assuming $\gamma \approx 0.5$. To summarize, the problem can be written as a following constrained nonlinear programming problem,

Problem 1-A:

$$\begin{aligned} \text{minimize} \quad & S(\boldsymbol{H}^\gamma, \boldsymbol{P}^\gamma; w) \\ \text{subject to} \quad & \boldsymbol{H}^{2\gamma} + \boldsymbol{P}^{2\gamma} = \boldsymbol{Y}^{2\gamma}, \\ & \boldsymbol{H}^\gamma \geq 0, \quad \boldsymbol{P}^\gamma \geq 0. \end{aligned}$$

Another reasonable setting is $\xi = \gamma \approx 0.5$, assuming 1-(c). In this case the problem is written as follows.

Problem 1-B:

$$\begin{aligned} \text{minimize} \quad & S(\boldsymbol{H}^\gamma, \boldsymbol{P}^\gamma; w) \\ \text{subject to} \quad & \boldsymbol{H}^{\gamma} + \boldsymbol{P}^{\gamma} = \boldsymbol{Y}^{\gamma}, \\ & \boldsymbol{H}^\gamma \geq 0, \quad \boldsymbol{P}^\gamma \geq 0. \end{aligned}$$

Explicit procedures to obtain approximate solutions to these optimization problems shall be described in Section 6.4.

6.3.2 Formulation 2: Considering Reconstructivity as a Cost Function

Aside from Problems 1-A and 1-B, we may also consider another approach, making some allowances for the difference between $\boldsymbol{H}^\xi + \boldsymbol{P}^\xi$ and $\boldsymbol{Y}^\xi$. In this subsection, instead of laying the strict constraint, we consider adding another cost term on the difference between them to the objective function, and derive an algorithm that minimizes thus obtained congregative objective function.

Although there are many possible distance measure between $\boldsymbol{H}^{\xi} + \boldsymbol{P}^{\xi}$ and $\boldsymbol{Y}^{\xi}$, we considered generalized Kullback-Leibler (KL) divergence, which is one of the basic statistical criterion that has been used in many fields including non-negative matrix factorization (NMF) [17], [90], in order to measure the "distance" between two distributions. Hereafter we just call it KL divergence. The KL divergence of $\boldsymbol{Y}^{\xi}$ from $\boldsymbol{H}^{\xi} + \boldsymbol{P}^{\xi}$ is defined by

$$D_{\mathsf{KL}}(\boldsymbol{Y}^{\xi} \| \boldsymbol{H}^{\xi} + \boldsymbol{P}^{\xi}) := \sum_{n=0}^{N-1} \sum_{k=0}^{K-1} \left\{ Y_{n,k}^{\xi} \ln \frac{Y_{n,k}^{\xi}}{H_{n,k}^{\xi} + P_{n,k}^{\xi}} - Y_{n,k}^{\xi} + H_{n,k}^{\xi} + P_{n,k}^{\xi} \right\}, \tag{6.15}$$

where ξ is an exponential factor, which should be $\xi = 2\gamma$ because of the requirement for the homogeneity of the objective function (see Appendix B). Using KL divergence, a relaxed optimization problem is defined as follows,

Problem 2:

$$\begin{aligned} \text{minimize} \quad & S(\boldsymbol{H}^{\gamma}, \boldsymbol{P}^{\gamma}; w) \\ & \quad +\mu D_{\mathsf{KL}}(\boldsymbol{Y}^{2\gamma}, \boldsymbol{H}^{2\gamma} + \boldsymbol{P}^{2\gamma}) \\ & =: U(\boldsymbol{H}^{\gamma}, \boldsymbol{P}^{\gamma}; \boldsymbol{Y}^{\gamma}, w, \mu) \\ \text{subject to} \quad & \boldsymbol{H}^{\gamma} \geq 0, \quad \boldsymbol{P}^{\gamma} \geq 0 \end{aligned}$$

where μ is a weight constant. The problem is identical to Ono's previous study [112] when $N' = K' = 1$. An algorithm to solve this optimization problem shall be shown in the next section.

6.4 Derivation of the Algorithms

In this section, we consider deriving algorithms that give practical solutions to the three problems described above. The algorithms are based on iterative updating, that gives a sequence of $(\boldsymbol{H}^{\gamma}, \boldsymbol{P}^{\gamma})$ which *decrease* (to be precise, *does not increase*) the objective function S or U, satisfying the constraints.

6.4.1 Optimization algorithm for Problem 1-A

Tentatively, let us ignore the non-negativity constraint for convenience. This constraint shall be worked out later. Besides, let us concentrate on a single bin (n, k) of all the NK

bins to make the discussion simpler. Now we have a following subproblem.

$$\begin{aligned} \text{minimize} \quad & S({H_{n,k}}^{\gamma}, {P_{n,k}}^{\gamma} | \boldsymbol{H}^{\gamma}, \boldsymbol{P}^{\gamma}; w) \\ \text{subject to} \quad & H_{n,k}^{2\gamma} + P_{n,k}^{2\gamma} = Y_{n,k}^{2\gamma}. \end{aligned}$$

Let us solve the problem on the basis of the standard procedure of Lagrange multiplier method. Lagrangian function is given by

$$\mathcal{L} := S({H_{n,k}}^{\gamma}, {P_{n,k}}^{\gamma} | \dots) + \lambda(H_{n,k}^{2\gamma} + P_{n,k}^{2\gamma} - Y_{n,k}^{2\gamma}), \tag{6.16}$$

where λ is a Lagrange multiplier. Solving the equations on the extrema of $\mathcal{L}$, i.e., $\partial\mathcal{L}/\partial({H_{n,k}}^{\gamma}) = \partial\mathcal{L}/\partial({P_{n,k}}^{\gamma}) = 0$, the equations on the stationary points are obtained as follows,

$$\begin{aligned} {H_{n,k}}^{\gamma} &= (2+\lambda)^{-1} H_{n,k}^{(n\text{-mean})}, & (6.17) \\ {P_{n,k}}^{\gamma} &= (2w+\lambda)^{-1} P_{n,k}^{(k\text{-mean})}, & (6.18) \end{aligned}$$

where

$$\begin{aligned} H_{n,k}^{(n\text{-mean})} &:= \frac{1}{2N'} \sum_{n'=1}^{N'} ({H_{n+n',k}}^{\gamma} + {H_{n-n',k}}^{\gamma}), & (6.19) \\ P_{n,k}^{(k\text{-mean})} &:= \frac{1}{2K'} \sum_{k'=1}^{K'} ({P_{n,k+k'}}^{\gamma} + {P_{n,k-k'}}^{\gamma}), & (6.20) \end{aligned}$$

which indicate the moving averages of the time-frequency bins around (n,k), excluding (n,k) itself. By substituting ${H_{n,k}}^{\gamma}$ and ${P_{n,k}}^{\gamma}$ in $\partial\mathcal{L}/\partial\lambda = 0$ by (6.17) and (6.18), a quartic equation on λ is derived as follows,

$$\left(\frac{H_{n,k}^{(n\text{-mean})}}{2+\lambda}\right)^2 + \left(\frac{P_{n,k}^{(k\text{-mean})}}{2w+\lambda}\right)^2 = Y_{n,k}^{2\gamma}. \tag{6.21}$$

This equation, however, is not easily solved practically for general w, as it is a quartic equation on λ. Nevertheless, assuming that $w = 1$, the equation becomes a quadratic equation on λ as follows,

$$(\lambda+2)^2 = \frac{1}{Y_{n,k}^{2\gamma}} \left\{ (H_{n,k}^{(n\text{-mean})})^2 + (P_{n,k}^{(k\text{-mean})})^2 \right\}. \tag{6.22}$$

Using thus obtained λ, and noting that ${H_{n,k}}^{\gamma}$ and ${P_{n,k}}^{\gamma}$ should be positive, the equations on extrema are derived as follows,

$$\begin{aligned} {H_{n,k}}^{\gamma} &= \frac{H_{n,k}^{(n\text{-mean})}}{\sqrt{(H_{n,k}^{(n\text{-mean})})^2 + (P_{n,k}^{(k\text{-mean})})^2}} {Y_{n,k}}^{\gamma} & (6.23) \\ {P_{n,k}}^{\gamma} &= \frac{P_{n,k}^{(k\text{-mean})}}{\sqrt{(H_{n,k}^{(n\text{-mean})})^2 + (P_{n,k}^{(k\text{-mean})})^2}} {Y_{n,k}}^{\gamma}. & (6.24) \end{aligned}$$

Algorithm 6.3 HPSS 1-A

```
Given a complex spectrogram $\hat{\boldsymbol{Y}} \in \mathbb{C}^{N\times K}$, and take its absolute value $\boldsymbol{Y} = |\hat{\boldsymbol{Y}}| \in \mathbb{R}^{N\times K}$.
Set initial values $\boldsymbol{H}^\gamma$ and $\boldsymbol{P}^\gamma$. We simply set them as $\boldsymbol{H}^\gamma = \boldsymbol{P}^\gamma = \boldsymbol{Y}^\gamma/\sqrt{2}$ in this thesis.
Update $\boldsymbol{H}^\gamma$ and $\boldsymbol{P}^\gamma$ using (6.25) and (6.26).
Iterate iii for $I$ times.
Apply inverse STFT in order to obtain audible waveforms $h(t)$ and $p(t)$ using $\boldsymbol{H}, \boldsymbol{P}$ and phase spectrogram of $\hat{\boldsymbol{Y}}$, i.e., $x(t) = \mathsf{STFT}^{-1}[\boldsymbol{X}\hat{\boldsymbol{Y}}/\boldsymbol{Y}]$.
```

We can use (6.23) and (6.24) as a tentative solution for a single time-frequency bin (n, k). That is,

$$H_{n,k}{}^\gamma \leftarrow \text{r.h.s. of (6.23)}, \tag{6.25}$$

$$P_{n,k}{}^\gamma \leftarrow \text{r.h.s. of (6.24)}. \tag{6.26}$$

This substitution *decreases* (precisely, *does not increase*) the objective function S. Moreover, evidently, applying the discussion above orderly for all time-frequency bins never increase S, too.[⋆1] By summarizing the discussion above and filling in with details, the whole procedure is written as shown in Algorithm 6.3.

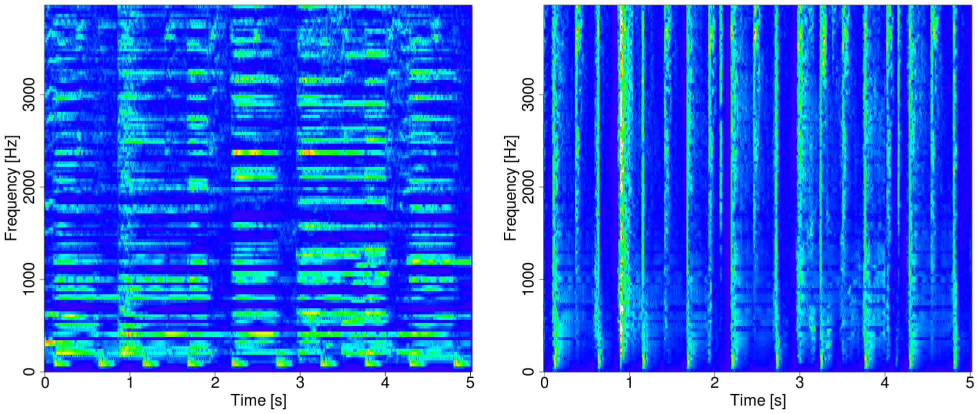
Note, despite the convexity of the objective function S (see Appendix), the problem is not convex programming, because the equality constraint does not satisfy the requirement of convex programming that it should be affine. Therefore, just decreasing objective function does not necessarily result in global minimum. Then we additionally need to consider a heuristics. The initial value in line 2, Algorithm 6.3, indeed, is a reasonable heuristics which has been empirically effective in our experiments.

Figure 6.4 shows an example of the result of Method 1-A. It is observed that the vertical components in $\boldsymbol{H}$ and the horizontal components in $\boldsymbol{P}$ are smoothed out in (b).

[⋆1] The formula are quite similar to the FitzGedald's median filtering [32]. Indeed, we may obtain the FitzGedald's method just by replacing ($\cdot$-mean) by ($\cdot$-median) in (6.26), where

$$H_{n,k}^{(n\text{-}\mathsf{median})} := \text{median}((H_{n+n',k})_{-M\le n'\le M}), P_{n,k}^{(k\text{-}\mathsf{median})} := \text{median}((P_{n,k+k'})_{-M\le k'\le M}), \tag{6.27}$$

Given a real-valued data set $(x_1, x_2, \ldots, x_n)$, it is known that the average $m = \sum_{i=1}^n x_i/n$ minimizes the square error $\sum_{i=1}^n (x_i - m)^2$, (i.e., 2-norm-based cost), while a median $m = \text{median}((x_i)_{1\le i\le n})$ minimizes 1-norm-based cost $\sum_{i=1}^n |x_i - m|$.



(a) (b)

Figure 6.4:: An example of the HPSS result of Method 1-A. Original input spectrogram $\boldsymbol{Y}$ was Figure 6.1(a). (a) and (b) are resultant spectrograms; (a) is harmonic and (b) is percussive. Number of Iteration was $I = 5$, and the parameter is $N' = K' = 8$.

6.4.2 Optimization algorithm for Problem 1-B

We may similarly derive the optimization procedure for Problem 1-B. The updating formulae are written as follows[⋆2].

$$H_{n,k}{}^{\gamma} \leftarrow \rho(\alpha_{n,k}; 0, Y_{n,k}{}^{\gamma}) \tag{6.28}$$

$$P_{n,k}{}^{\gamma} \leftarrow \rho(\beta_{n,k}; 0, Y_{n,k}{}^{\gamma}) \tag{6.29}$$

where

$$\alpha_{n,k} = \frac{1}{2}\left(Y_{n,k}{}^{\gamma} + H_{n,k}^{(n\text{-mean})} - P_{n,k}^{(k\text{-mean})}\right) \tag{6.30}$$

$$\beta_{n,k} = \frac{1}{2}\left(Y_{n,k}{}^{\gamma} - H_{n,k}^{(n\text{-mean})} + P_{n,k}^{(k\text{-mean})}\right) \tag{6.31}$$

$$\rho(x; l, u) = \begin{cases} l & \text{if } x < l \\ x & \text{if } l \leq x \leq u \\ u & \text{if } u < x. \end{cases} \tag{6.32}$$

This substitution *decreases* (precisely, *does not increase*) the objective function S by the same reason above. Thus whole the procedure is written as Algorithm 6.5. Note, Problem 1-B is a convex programming, which implies that a local optimum is always a global optimum.

[⋆2] We assumed $w = 1$ for convenience, but we may also easily derive the updating formula for general $w > 0$.

Algorithm 6.5 HPSS 1-B

```
1: Given a signal $y(t)$.
2: Calculate complex spectrogram $\hat{\boldsymbol{Y}} = \mathsf{STFT}(L)\,[y(t)] \in \mathbb{C}^{N\times K}$, using a reasonable frame $L$.
3: Take its absolute value $\boldsymbol{Y} = |\hat{\boldsymbol{Y}}| \in \mathbb{R}^{N\times K}$
4: Set initial values $\boldsymbol{H}^\gamma$ and $\boldsymbol{P}^\gamma$. We simply set them as $\boldsymbol{H}^\gamma = \boldsymbol{P}^\gamma = \boldsymbol{Y}^\gamma/\sqrt{2}$. (See also Appendix B)
5: for  $I$ times do
6:     Update $\boldsymbol{H}^\gamma$ and $\boldsymbol{P}^\gamma$ using (6.25) and (6.26).
7: end for
8: Apply inverse STFT in order to obtain audible waveforms $h(t)$ and $p(t)$ using $\boldsymbol{H}, \boldsymbol{P}$ and phase spectrogram of $\hat{\boldsymbol{Y}}$, i.e., $h(t) = \mathsf{STFT}^{-1}(L)[\boldsymbol{H}\hat{\boldsymbol{Y}}/\boldsymbol{Y}]$. etc.
```

6.4.3 Optimization algorithm for Problem 2

Similarly to Problem 1-A and 1-B, we consider this problem element by element. Let us consider the derivatives of the objective function U w.r.t. a variable under consideration. The solution that minimize $U({H_{n,k}}^\gamma|\boldsymbol{H}^\gamma, \boldsymbol{P}, \boldsymbol{Y})$ w.r.t. the variable ${H_{n,k}}^\gamma$ should hold the following equation

$$\frac{\partial U}{\partial({H_{n,k}}^\gamma)} = 0. \tag{6.33}$$

Therefore, applying the updating formulae that are derived by solving (6.33) does not increase U. It is not difficult to solve this equation theoretically, but inconveniently, (6.33) result in cubic equations on ${H_{n,k}}^\gamma$, which is costly to solve. We then consider simplification of the problem, using the following trick, which reduces the order of the problem into quadratic. The idea of following discussion is based on the techniques that were used in some other studies such as [68].

What is causing the cubic equation is the addition in the denominator of $\ln\{Y_{n,k}^{2\gamma}/(H_{n,k}^{2\gamma} + P_{n,k}^{2\gamma})\}$, the first term of KL divergence. In order to remove the inconvenience, we factorize the KL divergence into two KL divergences, using a parameter $\theta_{n,k}$, $(0 \le \theta_{n,k} \le 1)$ as follows,

$$D_{\mathsf{KL}}(\boldsymbol{Y}^{2\gamma}\|\boldsymbol{H}^{2\gamma} + \boldsymbol{P}^{2\gamma}) \le D_{\mathsf{KL}}(\boldsymbol{\theta}\boldsymbol{Y}^{2\gamma}\|\boldsymbol{H}^{2\gamma}) + D_{\mathsf{KL}}((\boldsymbol{1}-\boldsymbol{\theta})\boldsymbol{Y}^{2\gamma}\|\boldsymbol{P}^{2\gamma}). \tag{6.34}$$

This inequality is easily proved using an inequality $-\ln(x+y) \le -\theta\ln(x/\theta) - (1-\theta)\ln(y/(1-\theta))$, where $x, y > 0, 0 < \theta < 1$. The equality of (6.34) is satisfied only when

$$\theta_{n,k} = \frac{H_{n,k}^{2\gamma}}{H_{n,k}^{2\gamma} + P_{n,k}^{2\gamma}}. \tag{6.35}$$

Notes on Section 6.4.3: The derivation procedure of this section was not author's contribution, but was borrowed from the papers [112], and [146, Appendix].

The inequality (6.34) yields the following inequality,

$$U \leq S(\boldsymbol{H}^{\gamma}, \boldsymbol{P}^{\gamma}; w) + \mu\{D_{\mathsf{KL}}(\boldsymbol{\theta}\boldsymbol{Y}^{2\gamma} \| \boldsymbol{H}^{2\gamma}) + D_{\mathsf{KL}}((\boldsymbol{1}-\boldsymbol{\theta})\boldsymbol{Y}^{2\gamma} \| \boldsymbol{P}^{2\gamma})\}$$
$$=: U^{+}(\boldsymbol{H}^{\gamma}, \boldsymbol{P}^{\gamma}, \boldsymbol{\theta}; \boldsymbol{Y}^{\gamma}, w, \mu), \tag{6.36}$$

where U^+ is an auxiliary function that gives an upper bound of U. From here, we tentatively consider derivation of a sequence that decrease U^+ instead of U. The original purpose to obtain a sequence that decrease U shall be achieved in the next paragraph. The partial derivative of the auxiliary function $\partial U^+/\partial({H_{n,k}}^{\gamma}) = \partial U^+/\partial({P_{n,k}}^{\gamma}) = 0$, conveniently result in the following quadratic equations,

$$a_1({H_{n,k}}^{\gamma})^2 - 2b_1 {H_{n,k}}^{\gamma} - c_1 = 0, \quad a_2({P_{n,k}}^{\gamma})^2 - 2b_2 {P_{n,k}}^{\gamma} - c_2 = 0, \tag{6.37}$$

where

$$a_1 = 2 + \mu, \quad b_1 = H_{n,k}^{(n\text{-}\mathsf{mean})}, \quad c_1 = \mu\theta_{n,k} Y_{n,k}^{2\gamma},$$
$$a_2 = 2 + \mu', \quad b_2 = P_{n,k}^{(k\text{-}\mathsf{mean})}, \quad c_2 = \mu'(1-\theta_{n,k})Y_{n,k}^{2\gamma}, \quad \mu' = w^{-1}\mu.$$

Solving the quadratic equations on ${H_{n,k}}^{\gamma}$ and ${P_{n,k}}^{\gamma}$, noting that the solutions should be non-negative as well as the minimum of U^+, following updating formulae are obtained[⋆3],

$${H_{n,k}}^{\gamma} \leftarrow \frac{b_1 + \sqrt{b_1^2 + a_1 c_1}}{a_1},, \tag{6.38}$$

$${P_{n,k}}^{\gamma} \leftarrow \frac{b_2 + \sqrt{b_2^2 + a_2 c_2}}{a_2}. \tag{6.39}$$

In addition to (6.38) and (6.39), we can consider minimization of U^+ w.r.t. $\theta_{n,k}$. Clearly the $\theta_{n,k}$ that makes U^+ minimal is none other than (6.35), because

$$U^+(\theta_{n,k} = \text{r.h.s. of (6.35)}) = U \leq U^+(\theta_{n,k}). \tag{6.40}$$

(Note variables except $\theta_{n,k}$ are fixed here.) Therefore, substituting $\theta_{n,k}$ by the r.h.s of (6.35) also *decreases* (*does not increase*) U^+.

In the updating procedure (6.39) and (6.35), the auxiliary function U^+ does not increase. In addition, noting that $U^+ = U$ is satisfied just after updating $\theta_{n,k}$, it is verified that U also does not increase in the procedure. By summarizing the discussion above and filling in with details, the whole procedure is written as Algorithm 6.6.

[⋆3] Note these equations are identical to the original paper [112], when $w = 1, \mu = 2\sigma_{\mathrm{H}}^2 = 2\sigma_{\mathrm{P}}^2, N' = K' = 1$.

Algorithm 6.6 HPSS 2

```
1: Given a complex spectrogram $\hat{\boldsymbol{Y}}$, and take $\boldsymbol{Y} = |\hat{\boldsymbol{Y}}|$
2: Set initial values to $\boldsymbol{H}^\gamma = \boldsymbol{P}^\gamma = \boldsymbol{Y}^\gamma/\sqrt{2}$.
3: Update $\boldsymbol{H}^\gamma, \boldsymbol{P}^\gamma$ and $\boldsymbol{\theta}$ using (6.39) and (6.35).
4: Iterate iii for $I$ times.
5: Apply some postprocessings to $\boldsymbol{H}$ and $\boldsymbol{P}$. We applied Wiener mask in this thesis.
6: Apply inverse STFT in order to obtain $h(t)$ and $p(t)$ similarly to Problem 1.
```

Table 6.7: Computation time of single updating (*ESTIMATED* by averaging 100 times iterations). The length of input signal is 20 [s]. Unit: [ms].

	$M(:= N' = K')$									
Method	1	2	3	4	5	6	7	8	10	16
HPSS 1-A	28.5	29.5	39.1	36.0	41.5	41.8	43.6	40.8	48.5	57.3
HPSS 1-B	12.1	12.1	22.3	15.3	27.7	25.4	25.6	23.2	28.7	37.9
HPSS 2	41.7	43.1	51.3	50.4	54.9	57.3	57.2	55.3	60.5	66.8

Note, since we did not lay strict constraint on the distance between $\boldsymbol{H}+\boldsymbol{P}$ and $\boldsymbol{Y}$, the sum of separated signals are sometimes too distant from the original spectrogram. Therefore, we heuristically applied Wiener masking as a postprocessing after the iterations to modify these errors (5 in the above procedure). Nevertheless, the postprocessing is not altogether *ad hoc* in a sense that the updating equations approach asymptotically to Wiener masking when $\mu \to \infty$. If we interpreted μ to be a penalty factor, the postprocessing can be understood as a limit case of penalty function method.

6.5 Dependencies of Performance on $M := N' = K'$

In this section, we specify the value of N' and K' in S by conducting preliminary experiments. For simplicity we consider the case $N' = K' =: M$.

6.5.1 Evaluation of Computation Time of Single Update

We first evaluated the computation time of applying updating formulae of HPSS once. We used a 16 kHz sampled monaural audio signal of length 20 [s] as a sample data. The frame length was $L = 1024/16000$ [s], and the frame shift was $L/4$. The computer we used in the experiments was a laptop workstation DELL PRECISION M4500, Intel®Core$^{\text{TM}}$i7 CPU Q 740 @ 1.73GHz, and the OS was Linux. In evaluating the computation time, we applied the updating formulae 100 times, instead of applying them once, and we divided the measured computation time by 100.

Table 6.7 shows the computation time of each HPSS algorithm's single update, for $M = 1, 2, \ldots, 16$. It is observed that the computation time of each update is not proportional to M, but it is much faster. This is partly because of the reason that M affects only the computation cost of $H_{n,k}^{(n\text{-mean})}$ and $P_{n,k}^{(k\text{-mean})}$, and other operations such as square root, which is much more costly than computation of $H_{n,k}^{(n\text{-mean})}$, is independent of M. We may also observe in the table that the computation time of the case $M = 4$ is faster than $M = 3$, which contradict the fact that the calculation amount of the single update of HPSS is $O(NK \times (M + \text{const}))$. Nevertheless, this reverse phenomenon may not be an error, but is possibly attributable to the lower-level reasons of computer architecture, etc.

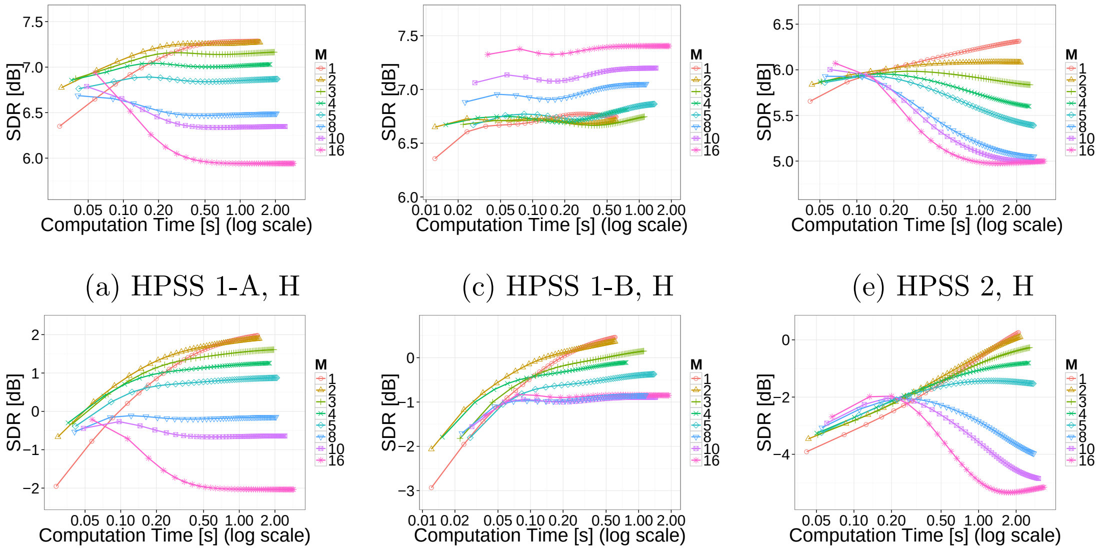
6.5.2 Examples of SDR improvement in HPSS updating

We conducted an experiment to see the relation between computation time and SDR improvement for each parameter, namely $M := N' = K'$, in order to decide M and the number of iteration I.

The data we used here were "Dreams" in BASS-dB and "Sunrise" in QUASI dataset. The length was 20 [s], and the sampling rate was 16 kHz. We mixed them in 5 dB (Harmonic to Percussive ratio). The computation time were not measured directly, but were estimated from the time that was evaluated in the previous subsection.

Figure 6.8 shows the result: the SDR [155] improvement curves of the harmonic and percussive components for each M. We may observe that M around 2 to 5 improve SDR faster than $M = 1$ in the beginning, but $M = 1$ catches up with and overtake others later, in many cases. It is also observed that too large M result in poorer performance in most cases.

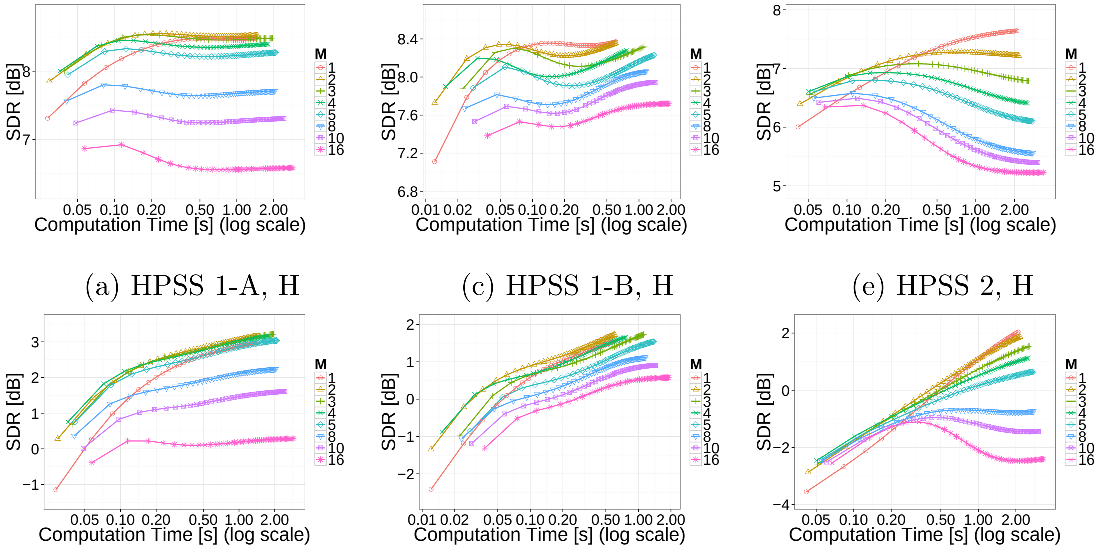
From these figures we may decide which M and iteration number is the best in terms of SDR. Considering the trade-off between the sound quality (SDR) and computation time, it would be reasonable to set $M = N' = K'$ and I as the values (a) and (b) in Table 6.9, 6.10, and 6.11. Of course, we may also consider other values for specific applications, but we only consider the two representatives for simplicity.



(a) HPSS 1-A, H (c) HPSS 1-B, H (e) HPSS 2, H

(b) HPSS 1-A, P (d) HPSS 1-B, P (f) HPSS 2, P

Another Dreamer – “Dreams” from BASS-dB dataset



(a) HPSS 1-A, H (c) HPSS 1-B, H (e) HPSS 2, H

(b) HPSS 1-A, P (d) HPSS 1-B, P (f) HPSS 2, P

Shannon Hurley – “Sunrise” from QUASI dataset

Figure 6.8:: Relation to computation time and SDR of the output Harmonic and Percussive signals, obtained by each HPSS. (a), (b) HPSS 1-A, (c), (d) HPSS 1-B, and (e), (f) HPSS 2. M denotes $M := N' = K'$. Each point indicates the ending time of the single iteration. Note computation time was not directly evaluated, but calculated from the results of Section 6.5.1 (Table 6.7).

Table 6.9: Parameter settings of HPSS 1-A

Parameter	Value (a)	Value (b)
Range M	1	2
Iteration I	20	10
Exponential γ	0.5	0.5
Frame L	1024 (64 [ms])	1024 (64 [ms])
Shift s	256 (16 [ms])	256 (16 [ms])
Window	hanning	hanning

Table 6.10: Parameter settings of HPSS 1-B

Parameter	Value (a)	Value (b)
Range M	1	2
Iteration I	15	5
Exponential γ	0.5	0.5
Frame L	1024 (64 [ms])	1024 (64 [ms])
Shift s	256 (16 [ms])	256 (16 [ms])
Window	hanning	hanning

Table 6.11: Parameter settings of HPSS 2 (The parameters w, μ and γ are basically based on [112].)

Parameter	Value (a)	Value (b)
Range M	1	2
Iteration I	30	10
Weight w	1	1
Weight μ	0.1	0.1
Exponential γ	1	1
Frame L	1024 (64 [ms])	1024 (64 [ms])
Shift s	256 (16 [ms])	256 (16 [ms])
Window	hanning	hanning
Post process.	Wiener mask	Wiener mask

Table 6.12: Parameter Settings of Median-Filtering-based method [32].

Parameter	Value (a)	Value (b)
Range M	8	4
Frame L	1024 (64 [ms])	1024 (64 [ms])
Shift$s$	256 (16 [ms])	256 (16 [ms])
Window	hanning	hanning
Post process.	Wiener mask	Wiener mask

Table 6.13: Parameter Settings of Open-BliSSART [136], [162].

Parameter	Value (a)	Value (b)
# of bases	30	50
Iteration	100 (default)	100
Window size	60 [ms] (default)	60 [ms]
Window overlap	30 [ms] (default)	30 [ms]
Window function	sine (default)	sine

6.6 Evaluation

6.6.1 Evaluations on Separation Performance to Relevant Techniques

Parameter Setting of HPSS

The parameters of HPSS are shown in Table 6.9, 6.10, 6.11. The parameters M, I were decided in the previous section, while others are empirically decided.

Comparative Methods

The methods we compared to were as follows. One was the latest version (release 1.2) of OpenBliSSART [136], [162][⋆4]. OpenBliSSART is an NMF-based general framework of audio source separation, which is designed laying weight on the efficiency of computation for the sake of the practical use. The parameters of OpenBliSSART were chosen on the basis of their recommendations described in the user's manual[⋆5] and demo files[⋆6]. Specifically, following commands were used,

```
(a) septool -v -c30 -l9 input.wav
(b) septool -v -c50 -l9 input.wav
```

where `-c30` and `-c50` represent that the number of NMF bases is (a) 30 and (b) 50. `-l9` represents the ID of training set, and ID 9 is the one for drum/harmonic separation. `-v` is a technical option not to overwrite the database of OpenBliSSART. The parameters of this method are shown in Table 6.13.

The other method was median-based harmonic/percussive sound separation proposed by FitzGerald [32], which is another extension of Ono's previous conference paper [112]. The parameters of this method are shown in Table 6.12. The parameter (a) is basically identical to the one described in their original paper [32], but not identical. In the original paper [32], the frame length was 92.8 [ms] = 4096/44100 [s], and the frame shift was its quarter. The

[⋆4] [Online (Visited Jan. 2014)] `http://openblissart.github.io/openBliSSART/`

[⋆5] [Online (Visited Jan. 2014)] `http://openblissart.github.io/openBliSSART/manual.pdf`

[⋆6] [Online (Visited Jan. 2014)] `https://github.com/openBliSSART/openBliSSART/blob/master/demo/README` difference comes from the difference of sampling rate. The parameter (b) is a referential one that may be faster than (a).

Dataset and Evaluation Criteria

The music signals that we used in experiment were excerpted from the following datasets:

(i) 6 pieces from MASS database [156]$^{\star 7}$. All of them are monaural, sampled at 16 kHz, and ≈ 10 [s] of length.

(ii) 8 of 11 songs from QUASI (QUaero Audio SIgnals) dataset [94], [153]$^{\star 8}$, excluding "Emily Hurst – Parting friends" which does not contain percussion, and two songs "Fort Minor – Remember the name" and "Vieux Farka Touré – Ana" that also appeared in MASS. The dataset is originally composed of separately recorded instruments and voices. We mixed them (to be specific, simply added the tracks) and trimmed 20 [s] from each song by ourselves.

(iii) 11 of 20 songs from BASS-dB [154] database$^{\star 9}$. We first removed 5 songs which do not suit for our purpose here. We then removed 4 of 15 songs which also appeared in QUASI dataset. Each song consists of separately recorded instruments and voices. In experiment we similarly mixed them and trimmed 20 [s] from each song by ourselves.

Thus we obtained 25 clips of length 10–20 [s]. In experiment, we mixed harmonic and percussive components in 5 and 10 dB (harmonic to percussive ratio). (Note, in real-world music, harmonic components are a little louder than percussive component in many cases.)

For evaluation criteria, we used SDR, signal to interference ratio (SIR), and signal to artifact ratio (SAR) [155]$^{\star 10}$ which are commonly used in many source separation tasks.We evaluated the values of both separated $\boldsymbol{H}$ components and $\boldsymbol{P}$ components.

[⋆7] [Online (Visited Oct. 2013)] `http://www.mtg.upf.edu/static/mass/resources/`

[⋆8] [Online (Visited Oct. 2013)]
`http://www.tsi.telecom-paristech.fr/aao/en/2012/03/12/Quasi`

[⋆9] [Online (Visited Oct. 2013)] `http://www.inria.fr/metiss/bass-db/`
[Online (Visited Oct. 2013)] `http://bass-db.gforge.inria.fr/BASS-dB/?show=browse%id=mtracks`.

[⋆10] We basically ran BSS_EVAL 3.0 [155] (written in MATLAB) on GNU Octave 3.2.3. Given n sources and n estimates, BSS_EVAL automatically identifies the best permutation of the correspondence between sources and estimates from $n!$ possible permutations on the basis of SIR before the evaluation. In this experiment, however, we deleted the permutation estimation subroutine from the original BSS_EVAL, because the source separation techniques are supposed not only to separate a signal but also to label their output signals either 'harmonic' or 'percussive' in this task, and therefore, there is no ambiguity of permutation.

Results and Discussion

Figure 6.14 shows SDR, SIR, and SAR of the separated harmonic and percussive signals. In Figure 6.14 (a), (b) and (c), it is observed that HPSS algorithms outperform, or if not, perform almost comparably to the others in terms of SDR and SIR, on average, while Figure 6.14 (d) shows that HPSS methods are not as effective as OpenBliSSART in terms of SIR of percussive components. Figure 6.14 (e) and (f) show the SAR of the resultant signals. HPSS 1-A and HPSS 2 (b) outperformed NMF-based method in terms of SAR of harmonic components, and all the HPSS methods outperformed NMF-based method in terms of SAR of percussive components, on average.

In summary, in this section we verified that the performances of our methods are by and large comparable to the others, one of which is based on a common frameworks of NMF-based music processing.

6.6.2 Evaluation of Computation Time

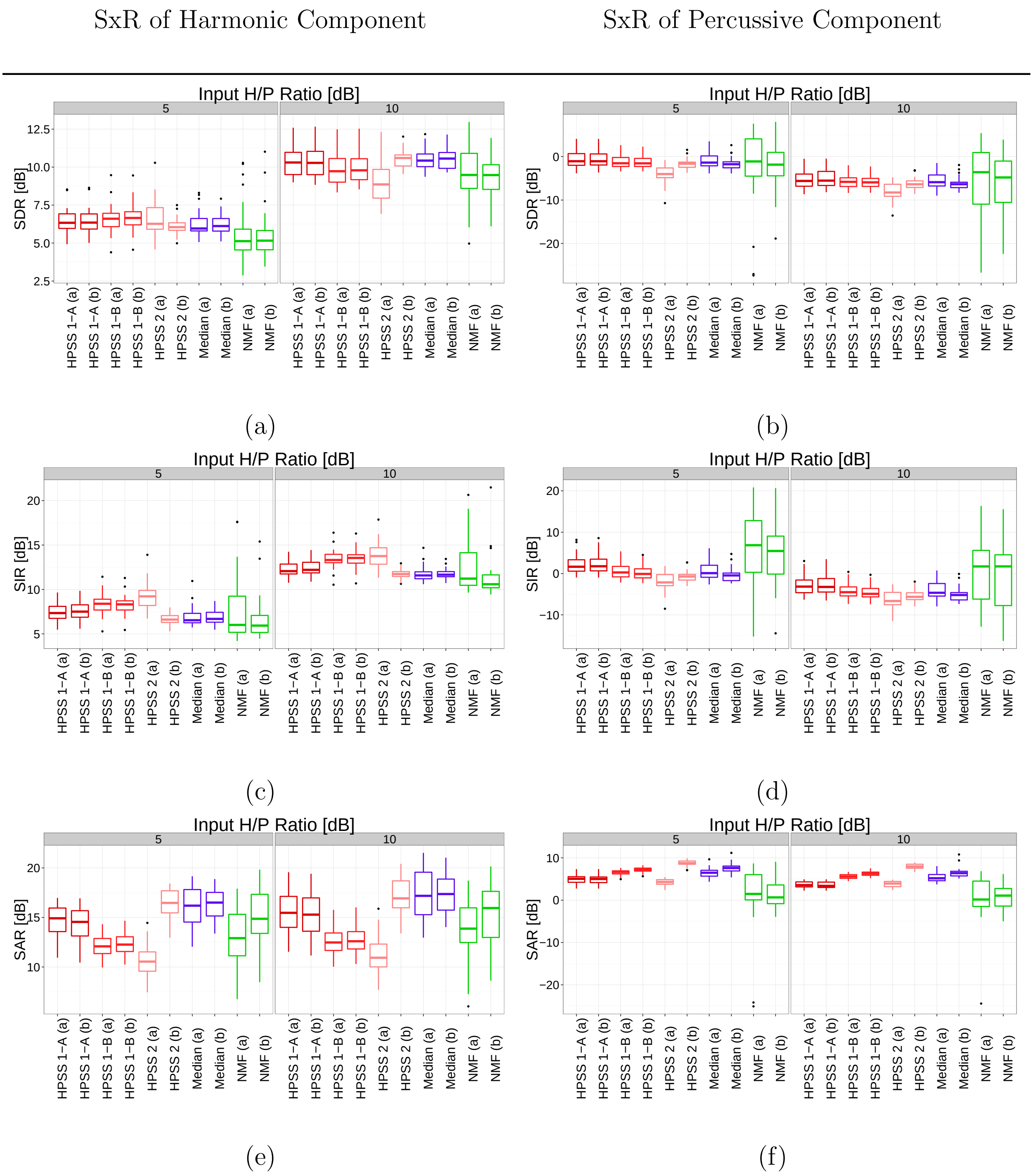
Practically, the computation cost is an important issue to use a technique in real world. In this section, we compared the computational efficiency of each method. The parameter of each method was the same as the previous section.

Experimental Condition

All of HPSS and FitzGedald's median filtering were implemented in C++ in a same framework by the authors. The type of all the variables was `double` (double floating point number.) In median filtering, we used the introsort algorithm `std::sort` in C++ standard library. In DFT, we used FFTW3 [40], which is one of the standards. We compiled all the programs using GNU C++ Compiler 4.6.3, with the optimization option `-O3`. The computer we used for evaluation was the same laptop workstation (DELL PRECISION M4500).

Results and Discussion

Table 6.15 shows the computation time and the real-time factor (RTF). It shows that proposal methods require quite short computation time. Indeed, HPSS 1-B (b) is the fastest among the all, and it can process three-minute song in 4 [s]. These digits indicate that the methods can process the data in real time. (Implementation of real-time HPSS is described in [112] and Chapter 4.)



(a)

(b)

(c)

(d)

(e)

(f)

Figure 6.14:: (a) Boxplot of SDR of harmonic components for 25 songs, (b) SDR of percussive components, (c) SIR of harmonic components, (d) SIR of percussive components, (e) SAR of harmonic components, (f) SAR of percussive components. "Median" indicates the method proposed by FitzGerald [32]. "NMF" indicates OpenBliSSART [136], [162].

Table 6.15: *TOTAL* computation time of each method to process 16-kHz sampled monaural music signals of length 1 minute and 3 minutes. The digits *INCLUDE* the computation time of STFT, Wiener masking, inverse STFT, and I/O.

	60-second song		180-second song	
Method	Time	RTF	Time	RTF
HPSS 1-A (a)	2.70 [s]	4.5×10^{-2}	9.9 [s]	5.5×10^{-2}
HPSS 1-A (b)	1.90 [s]	3.2×10^{-2}	6.5 [s]	3.6×10^{-2}
HPSS 1-B (a)	1.60 [s]	2.7×10^{-2}	5.2 [s]	2.9×10^{-2}
HPSS 1-B (b)	1.31 [s]	2.2×10^{-2}	4.0 [s]	2.2×10^{-2}
HPSS 2 (a)	4.87 [s]	8.1×10^{-2}	15.3 [s]	8.5×10^{-2}
HPSS 2 (b)	2.78 [s]	4.6×10^{-2}	8.6 [s]	4.8×10^{-2}
Median (a)	3.10 [s]	5.2×10^{-2}	9.9 [s]	5.5×10^{-2}
Median (b)	2.32 [s]	3.9×10^{-2}	7.0 [s]	3.9×10^{-2}
OpenBliSSART (a)	10.7 [s]	18×10^{-2}	28.6 [s]	16×10^{-2}
OpenBliSSART (b)	14.8 [s]	24×10^{-2}	42.5 [s]	24×10^{-2}

Table 6.16: Computation time of each method to process the songs, *EXCLUDING* STFT, Wiener masking, inverse STFT, and I/O. That is, the digits below are only on executing updating formulae of HPSS (or applying median filtering).

	60-second song		180-second song	
Method	Time	RTF	Time	RTF
HPSS 1-A (a)	1.61 [s]	2.7×10^{-2}	4.9 [s]	2.6×10^{-2}
HPSS 1-A (b)	0.89 [s]	1.5×10^{-2}	2.9 [s]	1.6×10^{-2}
HPSS 1-B (a)	0.60 [s]	1.0×10^{-2}	1.7 [s]	0.94×10^{-2}
HPSS 1-B (b)	0.22 [s]	0.37×10^{-2}	0.67 [s]	0.42×10^{-2}
HPSS 2 (a)	3.76 [s]	6.3×10^{-2}	11.3 [s]	6.3×10^{-2}
HPSS 2 (b)	1.48 [s]	2.5×10^{-2}	4.3 [s]	2.4×10^{-2}
Median (a)	1.25 [s]	2.1×10^{-2}	3.8 [s]	2.1×10^{-2}
Median (b)	0.56 [s]	0.93×10^{-2}	1.7 [s]	0.94×10^{-2}

Moreover, considering the computation time of the core of HPSS, i.e., HPSS updating formulae, excluding subsidiary processing such as STFT, I/O, etc, the efficiency outstands. It is shown in Table 6.16. Comparing the digits shown in Table 6.15 and 6.16, we may find that the computation cost of HPSS updating formula is much less than, or comparable to, other subsidiary processings. For example, HPSS 1-B (b) requires 4.0 [s] to process a 3-minute song, but only 0.67 [s] of whole the processing time is attributable to HPSS updating formulae, while the other 3.3 [s] is attributable to subsidiary processings such as STFT and I/O. This fact also supports the efficiency of HPSS.

Comparing to OpenBliSSART, which can be a representative of NMF-based methods, the efficiency especially outstands. In specific, comparing HPSS 1-B (b) and OpenBliSSART (b), HPSS is about 50 times faster than OpenBliSSART. This computation efficiency is an advantage of the proposal method compared to other existing methods.

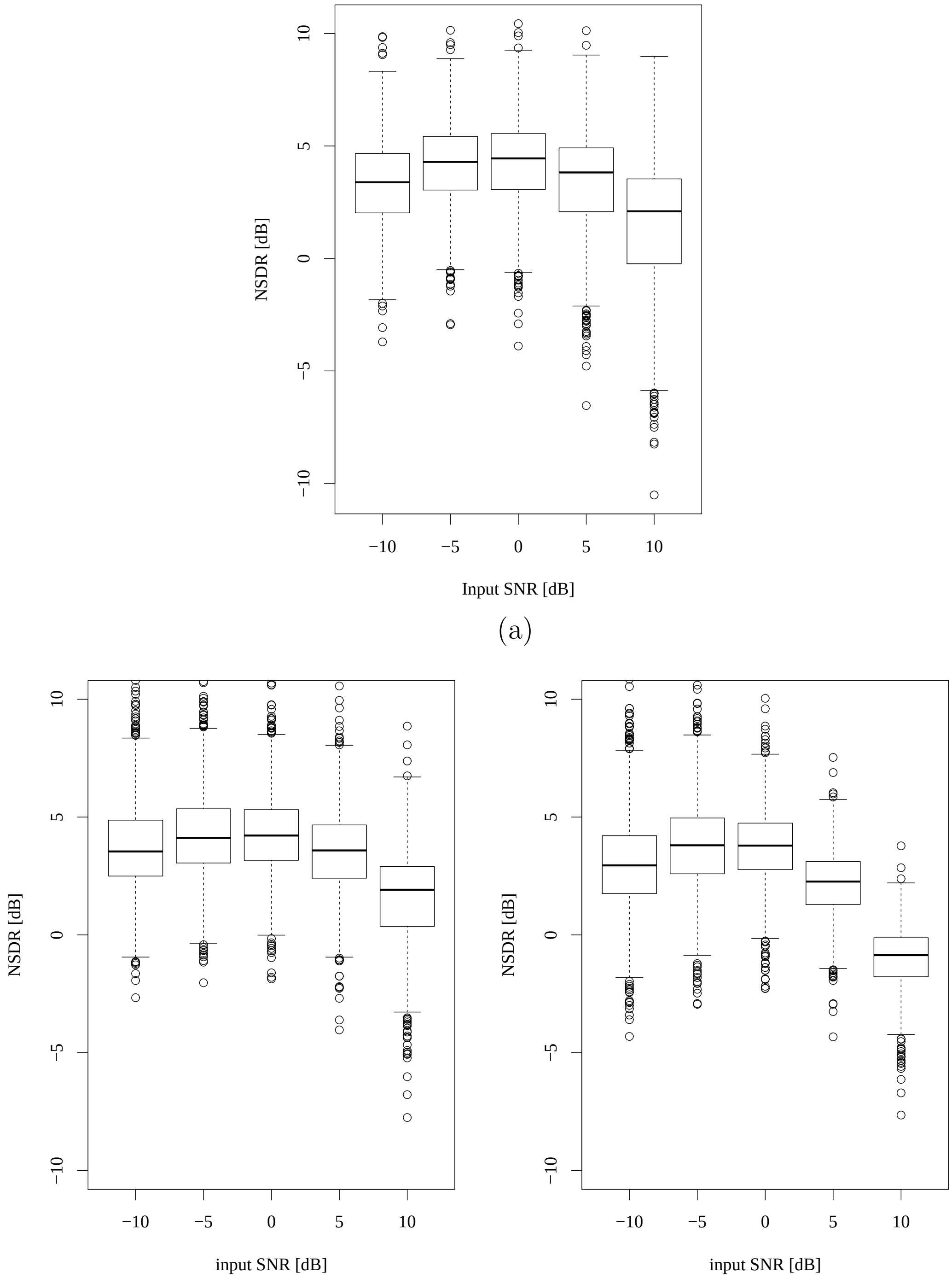
6.6.3 Application of the Modified HPSS to Singing Voice Enhancement in Chapter 2

We conducted the same experiment described in Chapter 2, replacing the conventional HPSS by modified HPSS 1-A, and 1-B, in order to show the effectiveness of the modified HPSS. The parameters are as follows. The frame length and shift of HPSS (short) and HPSS (long) were identical to Chapter 2. N' and K' were $N' = K' = 2$, and the number of iteration was 10.

Figure 6.17 shows the result. The figure shows that the separation performance in terms of NSDR is not much different from the result in Chapter 2 Nevertheless, as seen previously, the modified HPSS 1-A and 1-B are faster than HPSS 1 (a) which is identical to the conventional HPSS.

6.7 Summary of Chapter 6

In this chapter, we described the formulation of HPSS attempting some possibilities of reformulation: long-term relation and the strict reconstructivity. The methods to solve the modified problems are successfully obtained, and it is shown that the reformulated techniques have their advantages; long-term relation accelerated the performance in beginning several updates; strict reconstructivity simplified the formulation of HPSS, and it also reduced the computation cost as a consequence.



(a)

(b)

(c)

Figure 6.17:: (a) This figure is identical to Figure 2.10(a) in page 36. (b) Boxplot of NSDR of the proposed method for 1000 songs in MIR-1K dataset, using two-stage modified HPSS 1-A. (c) Boxplot of NSDR of the proposed method for 1000 songs in MIR-1K dataset, using two-stage modified HPSS 1-B.

We then applied the modified HPSS 1-A and HPSS 1-B to the two-stage HPSS similarly to Chapter 2. Experimental results showed that the performance was not much different, especially the case of HPSS 1-A. This shows that the modified HPSS processes the singing voice in shorter time without significant loss of the separation performance.

Chapter 7

Conclusion

7.1 Summary of the Thesis

This thesis described techniques on singing voices, which exploit spectral fluctuation. Fluctuation is one of the principal characteristics of singing voice, but has been sufficiently studied in the literature of singing voice enhancement. This thesis described a promising approach to the fluctuation of singing voice in this problem.

In order to capture the fluctuation of singing voice, we considered two differently-resolved spectrograms, one of which has rich temporal resolution and poor frequency resolution, while the other has the opposite resolution. The idea was that the behavior of fluctuating component is dependent on the time-frequency resolution of spectrogram, and a simple technique Harmonic/Percussive Sound Separation (HPSS) can separate them from two typical classes of instruments: harmonic (such as piano) and percussive.

On the basis of the idea, we proposed two-stage HPSS, as well as its application to an Music Information Retrieval (MIR) task and music application. We then considered the extension of the idea, and we also made some discussion the fundamental technique HPSS.

Each chapter described following.

Chapter 2

The technique described in Chapter 2, vocal extraction, is the principal contribution of this dissertation. In this dissertation, instead of modeling complicated properties of singing voice on an ordinary spectrogram, we considered to adjust the time-frequency resolution of the music spectrogram to be suitable for our purpose of singing voice enhancement.

To be specific, we considered two spectrograms that have different time-frequency resolution. On one of the two spectrograms, fluctuating components appear quite similarly to harmonic components, while on the other spectrogram, they appear quite similarly to percussions. Thus we may extract singing voice from the mixed music signals just by applying a simple technique HPSS twice, which we called two-stage HPSS.

Experimental evaluations showed that the performance of our approach is considerably better than the existing singing voice extraction techniques.

Chapter 3

This chapter described a straightforward application of two-stage HPSS described in Chapter 2 to a MIR problem, namely an audio melody extraction. Despite the simplicity of the pitch estimation technique itself, the performance of our melody extraction was better than other methods submitted to Music Information Retrieval Exchange (MIREX), in low-SNR (vocal to accompaniment ratio) conditions. The result of Chapter 3 proves the effectiveness of the singing voice enhancement technique shown in Chapter 2 as a preprocessing for an MIR task.

Chapter 4

Chapter 4 showed another application of the two-stage HPSS described in Chapter 2 as a singing voice suppression technique. Using two-stage HPSS as a singing voice suppressor, we proposed an audio-to-audio karaoke system. The system has not only vocal suppression but also a function of key transposition, which is a common function in MIDI-based karaoke systems. The whole system worked in real-time, which would be advantageous in the recent streaming-based music applications.

Chapter 5

This chapter described an extension of the concept of Chapter 2. In Chapter 2, we considered only two time-frequency resolutions, and applied HPSS on them. In contrast, in this chapter we considered more number of time-frequency resolutions and HPSS, to separate a signal by their characteristic time-scale of fluctuation. We called this procedure Concurrent Multiple HPSS (CMHPSS). By using the components separated by multiple

HPSS, we defined a novel audio feature, which we called Characteristic Fluctuation Timescale (CFTS). We applied thus obtained new feature to the speech/singing discrimination task, and verified that the discrimination accuracy was better than the case in which we only used Mel-frequency Cepstral Coefficients (MFCC), which is one of the most standard audio features. In addition, the joint use of MFCC and CFTS further improved the discrimination performance.

Chapter 6

Chapter 6 described HPSS in detail, independent of the discussion on singing voice. We also considered some possibilities of reformulation of HPSS in order to improve the performance of the techniques in the previous chapters, all of which are based on HPSS. We specifically considered long-term relation on spectrogram and strict reconstructivity. Experimental evaluation showed these modifications improved the computation time and performance of HPSS in some conditions. This modification did not distinctively improve the separation performance of two-stage HPSS, but the computation efficiency was improved without remarkable loss of the separation performance.

7.2 Future Work

The list of works remained in the future is as follows.

Sophistication of each Technique for each Purpose : Source Separation Techniques

HPSS still has rooms for improvement, because these methods are based only on “anisotropic smoothness” and a subsidiary constraint, “the sum of separated component should be equal to the original signal.” Another many properties may be effectively added in HPSS model, such as harmonicity of harmonic instruments, etc.

It is the same for the two-stage HPSS. In this dissertation we only exploited the fluctuation of singing voice, ignoring the harmonicity and timbrel features, which are also important characteristics of singing voice. For better quality of separation, we may need to incorporate these features into the methods in the future.

Another future work in source separation is the investigations on the robustness for some sound effects such as reverberations and nonlinear distortions that are sometimes observed in real-world musics.

Sophistication of each Technique for each Purpose : MIR and Applications

The method of Audio Melody Extraction (AME) discussed in Chapter 3 was one of the simplest one, thanks to the effectiveness of the preprocessing. It also has a room for improvement, such as considering melody-absence. In addition to AME, we will also attempt to apply two-stage HPSS to other MIR tasks related to singing voice, such as singer identification.

Our future study on karaoke applications will include a development of a novel karaoke application that incorporates "Euterpe" and an automatic accompaniment system, which follows the singing voice of the user, and plays the accompaniment in accordance with the user's singing voice, for example.

Joint Model to Global Musicologic Context

An important challenge in the future would be the connection of the use of global context of music, which seems to be an essential but difficult problem. In order to exploit global context of music, further intense studies would be required, to connect the signal processing layer and the symbolic layer.

Appendix A

Fundamental Knowledge on Music and Digital Signals

A.1 Fundamental Knowledge of Music

See also [83, §1.1], in which some fundamental knowledge on music required in music information processing are shown.

Unit System of Frequency Interval: Cent, Semitone, Octave and Others

In music the following terminology is used to express the intervals between two frequencies f_1, f_2. If the ratio satisfies $f_1/f_2 = 2^{n/1200}$, then the distance of f_1 from f_2 is called n cents. 100 cents is also called 1 semitone. 12 semitones, i.e., 1200 cents, is also called 1 octave. In other words, cent, semitone, and octave are units of the log frequency. 2 semitones is sometimes called wholetone. 1 semitone, 2 semitones, etc., are also called as "minor 2nd," "Major 2nd," etc. These terminology are dependent on the conventions.

MIDI Note Number and Conventional Note Names

The note number of MIDI is defined by the number of semitones from a note "C−1". The range of MIDI note number is $n \in \mathbb{N}, 0 \leq n < 128$. If a MIDI note number n satisfies

$$n \equiv 0, 2, 4, 5, 7, 9, 11 \mod 12 \tag{A.1}$$

then the note has the following conventional note names

"C," "D," "E," "F," "G," "A," "B,"

in English and

"do," "re," "mi," "fa," "sol," "la," "si"

in Romance languages, respectively. In order to specify the octaves, the quotient

$$O = \lfloor n/12 \rfloor - 1, \tag{A.2}$$

is used, and the notes are expressed by "note name + O." For example,

MIDI note number 69 = "A4."

The names for the notes which are not defined above, i.e., $n \equiv 1, 3, 6, 8, 10 \mod 12$, are expressed by using $\sharp$ (+1 semitone) and $\flat$ (−1 semitone). For example, MIDI note number 61 is "C$\sharp$4" as well as "D$\flat$4."

MIDI Note Number to Frequency

A relation between MIDI note number $n \in \mathbb{N}$ and the frequency $f(n) \in \mathbb{R}$ is written as

$$f(n) = f_{\mathrm{A4}} \times 2^{(n-69)/12}, \tag{A.3}$$

where f_{A4} [Hz] is basically 440 Hz. Note, we may naturally extend the range of MIDI note number to $\mathbb{R}$. Thus all the the frequencies have the corresponding MIDI note number, which is defined by the inverse of (A.3)

Temperament

Historically speaking, the values of $2^{(n-69)/12}$ were originally defined by "simple" rational numbers. Intuitively, the frequency of "E5" was $f_{\mathrm{A4}} \times 3/2 = 660$ Hz instead of $f_{\mathrm{A4}} \times 2^{7/12} \approx 659.255$ Hz. (For more precise discussion, see historical articles on musical tuning.) That is, historical semitones were not necessarily 100 cents. These definitions were mostly replaced by the modern definition, because of the convenience, but they are sometimes used. See also Appendix C.2.

A.2 Digital Signal Processing

This section shows the fundamentals of digital signal processing and time-frequency distribution.

A.2.1 Digital Signal and Discrete Fourier Transform

Let $x(t)$ be an $\mathbb{R}$-valued signal, where $t = \tau/f_\mathrm{s}$ [s], and $\tau \in \mathbb{Z}$ is discrete time, where f_s [Hz] is the sampling rate. The signal may be non-zero when $0 \leq t < T$, where T [s] is the length of the signal. Let us define $x(t) = 0$ when $t < 0, T \leq t$.

In an ordinary audio data of PCM format (`*.wav`), the value of $x(t)$ for specific t is expressed by an 8-bit or 16-bit integer (“`unsigned char`” and “`signed short`” in C/C++). The ranges of the value are

$$0 \leq x(t) \leq 2^8 - 1 \qquad \text{(8 bit)} \tag{A.4}$$

$$-2^{15} \leq x(t) \leq 2^{15} - 1. \quad \text{(16 bit)} \tag{A.5}$$

respectively, but we can easily convert a signal into $\mathbb{R}$-valued signal within a range

$$-1 \leq x(t) < 1. \tag{A.6}$$

by following transformation

$$x(t) \mapsto (x(t) - 2^7)/2^7, \tag{A.7}$$

$$x(t) \mapsto x(t)/2^{15}. \tag{A.8}$$

We basically consider the range of signals as (A.6). When we apply some processings to a signal, the value sometimes exceeds the range. In these cases, we normalize the signal by following formula,

$$x(t) \mapsto x(t)/\max_t x(t). \tag{A.9}$$

Discrete Fourier Transform (DFT)

Discrete Fourier transform (DFT), $\mathbb{R}^L \to \mathbb{C}^L$, of a short-term signal $x(\tau)$, $(\tau = 0, 1, 2, \ldots L-1)$ is defined as follows,

$$\tilde{x}(k) = \mathsf{DFT}_L[x(t)] := \sum_{\tau=0}^{L-1} x(t) \exp\{-2\pi\sqrt{-1}\tau k/L\}, \quad (k = 0, 1, 2, \ldots, L-1). \tag{A.10}$$

It is easily verified that $\tilde{x}(k) = \tilde{x}(L-k)^*$ (* denotes complex conjugate) unless $k = 0$, and also that the image of the mapping essentially lies in $\mathbb{R}^2 \times \mathbb{C}^{L/2-1}$ instead of $\mathbb{C}^L$, when L is even. Hereafter we assume L is always even for convenience. Inverse DFT is given by

$$x(\tau) = \mathsf{DFT}_L^{-1}[\tilde{x}(k)] := \frac{1}{L} \sum_{k=0}^{L-1} \tilde{x}(k) \exp\{2\pi\sqrt{-1}\tau k/L\}. \tag{A.11}$$

If L is a product of small primes, DFT is effectively executed by an algorithm called fast Fourier transform (FFT). (See also Appendix.C.2)

A.2.2 Time-frequency Distribution

STFT and Other Time-frequency Representations

Although there are many time-frequency representations, such as wavelet transform [19], Constant Q Transform (CQT) that shall be discussed later, and Wigner-Ville distribution [18, Ch.8], short-term Fourier transform (STFT) has following advantages.

- STFT is based on very local information of the signal. That is, STFT does not require information of signal, which is much distant from the time under consideration t_0. We may obtain STFT spectrogram around the time t_0 only by the short-term signal $x(t), t_0 - \Delta < t < t_0 + \Delta$.
- Computation of STFT is typically less costly than other methods because of its simplicity.
- The "cross-term components," which are often found in Wigner-Ville distribution, basically do not appear in STFT spectrograms.
- STFT has efficient inverse transform methods.

See also [18, Ch.7].

Time-frequency Resolution

The temporal resolution and the frequency resolution of a spectrogram are s/f_s [s] and f_s/L [Hz], respectively. Thus the product of them is always s/L, which is a constant, typically 1/2 or 1/4 in this dissertation. This fact forms the basis of the method which shall be discussed in Chapter 2.

Note this fact is sometimes referred to as the "uncertainty principle" of signal analysis, though its interpretation in relation to quantum physics is controversial. See some discussions in [18, Ch. 3, § 6.9, § 15.5]. Nonetheless, we do not step into the discussion on this topic any further in this dissertation.

Constant Q Transform

The frequency resolution of STFT in lower frequencies is not rich enough to formulate intuitively a method that extracts information from music signals. Therefore, instead of STFT, CQT is often used in the literature of music signal processing.

Let us first define Gabor function, which is a product of a sine wave of frequency f [Hz], i.e., $\exp(2\pi f\sqrt{-1}t)$ (where the unit of t is [s] for convenience), and a Gaussian function of mean 0, variance σ^2 [s^2], i.e., $(2\pi\sigma^2)^{-1/2}\exp(-t^2/2\sigma^2)$, as follows,

$$G(t; f, \sigma^2) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(2\pi f\sqrt{-1}t - \frac{t^2}{2\sigma^2}\right). \tag{A.12}$$

Let us rewrite Gabor function using a new parameter Q as follows.

$$G_Q(t; f) = \frac{1}{\sqrt{2\pi}(\pi^{-1}f/Q)} \exp\left(2\pi f\sqrt{-1}t - \frac{t^2}{2(\pi^{-1}f/Q)^2}\right). \tag{A.13}$$

where Q is coordinated with other variables by the following formula.

$$Q = \frac{f}{\pi\sigma^{-1}} \tag{A.14}$$

In CQT , it is defined that Q is a constant. In other words, Q is a proportionality factor between f and $(\pi\sigma^{-1})$, which is the standard deviation of the Gaussian in Fourier domain, i.e., const $\times \exp\{-f^2/2(\pi\sigma^{-1})^2\}$. The CQT of a signal $x(t)$ is defined by the convolution as follows,

$$X(t', f) = \sum_{t=-L/2}^{L/2-1} G_Q(t; f)^* x(t + t'). \tag{A.15}$$

However, the straightforward calculation based on this formula is not efficient. We may improve it by transforming the formula into Fourier domain, and calculate it on the frequency domain. For details, see [10], [11], [119, App. B] and [106, § II-D].

Appendix B

Remarks on the Formulation of HPSS

B.1 Non-Convexity of the Optimization Problems and Heuristic Countermeasures

B.1.1 Discussion on Non-convexity of the Methods

If an optimization problem is not convex, just descending the objective function does not necessarily result in the global optimal. In such problems, in order to obtain a solution of better quality, some heuristics are needed. This section makes some notes on the non-convexity of the problems.

Convexity of Smoothness Function S

A twice differentiable real function $f(x_1, x_2, \dots, x_n)$ is called convex if its Hessian matrix

$$H(f) := \left(\frac{\partial^2}{\partial x_i \partial x_j} f \right)_{ij} (x_1, \dots, x_n) \tag{B.1}$$

is positive semi-definite for all $(x_1, \dots, x_n) \in \mathbb{R}^n$. (See also e.g., Rockafeller's book [128, Thm 4.5].) If the objective function of a non-constrained programming is convex, the local optima are always the global optima, and hence, by decreasing the objective function a global optimal is obtained.

The smoothness function S is convex, because Hessian matrix of S is positive semidefinite, which is verified by the fact that it is a diagonally-dominant (d.d.) real symmetric matrix, and its diagonal components are positive.[⋆1] In fact,

$$H(S) = \begin{bmatrix} \left(\dfrac{\partial^2 S}{\partial(H_{n,k}{}^{\gamma})\partial(H_{n',k'}{}^{\gamma})}\right)_{n,k,n',k'} & \left(\dfrac{\partial^2 S}{\partial(H_{n,k}{}^{\gamma})\partial(P_{n',k'}{}^{\gamma})}\right)_{n,k,n',k'} \\ \left(\dfrac{\partial^2 S}{\partial(P_{n,k}{}^{\gamma})\partial(H_{n',k'}{}^{\gamma})}\right)_{n,k,n',k'} & \left(\dfrac{\partial^2 S}{\partial(P_{n,k}{}^{\gamma})\partial(P_{n',k'}{}^{\gamma})}\right)_{n,k,n',k'} \end{bmatrix}, \tag{B.2}$$

where

$$\frac{\partial^2 S}{\partial(H_{n,k}{}^{\gamma})\partial(H_{n',k'}{}^{\gamma})} = \begin{cases} 4 & \text{if } n = n', k = k' \\ -2/N' & \text{if } 0 < |n - n'| \leq N', k = k' \\ 0 & \text{otherwise} \end{cases} \tag{B.3}$$

$$\frac{\partial^2 S}{\partial(P_{n,k}{}^{\gamma})\partial(P_{n',k'}{}^{\gamma})} = \begin{cases} 4 & \text{if } n = n', k = k' \\ -2/K' & \text{if } 0 < |k - k'| \leq K', n = n' \\ 0 & \text{otherwise} \end{cases} \tag{B.4}$$

$$\frac{\partial^2 S}{\partial(H_{n,k}{}^{\gamma})\partial(P_{n',k'}{}^{\gamma})} = \frac{\partial^2 S}{\partial(P_{n,k}{}^{\gamma})\partial(H_{n',k'}{}^{\gamma})} = 0, \tag{B.5}$$

implies

$$\left|\frac{\partial^2}{\partial(H_{n,k}{}^{\gamma})^2} S\right| \geq \sum_{(n',k')\neq(n,k)} \left|\frac{\partial^2 S}{\partial(H_{n,k}{}^{\gamma})\partial(H_{n',k'}{}^{\gamma})}\right| + \sum_{(n',k')} \left|\frac{\partial^2 S}{\partial(H_{n,k}{}^{\gamma})\partial(P_{n',k'}{}^{\gamma})}\right|, \tag{B.6}$$

$$\left|\frac{\partial^2}{\partial(P_{n,k}{}^{\gamma})^2} S\right| \geq \sum_{(n',k')\neq(n,k)} \left|\frac{\partial^2 S}{\partial(P_{n,k}{}^{\gamma})\partial(P_{n',k'}{}^{\gamma})}\right| + \sum_{(n',k')} \left|\frac{\partial^2 S}{\partial(H_{n,k}{}^{\gamma})\partial(P_{n',k'}{}^{\gamma})}\right|. \tag{B.7}$$

Non-convexity of Objective Function U

Contrary to the convexity of S, the objective function U is not convex. because of the non-convexity of $D_{\mathsf{KL}}(z^2|x^2 + y^2)$. Note that, even if functions $f(x)$ and $g(x)$ are convex, it is not always true that $f(g(x))$ is convex. Indeed, the second derivative of the function,

$$\frac{d^2}{dx^2} f(g(x)) = \frac{d}{dx}\{f'(g(x))g'(x)\} = f''(g(x))g'(x)^2 + f'(g(x))g''(x) \tag{B.8}$$

[⋆0] A matrix $(a_{ij}) \in \mathbf{M}(n \times n, \mathbb{R})$ is d.d. if it satisfy the property $|a_{ii}| \geq \sum_{j \neq i} |a_{ij}|$ for all i. It is known that d.d. real symmetric matrix with non-negative diagonal entries is positive semidefinite. See [9]. See also some standard books on advanced matrix analysis, including Horn & Johnson [61, Thm. 6.1.10, Cor. 7.2.3, etc], etc, which contains some discussions on *strictly* d.d. matrices (replace '$\geq$' above by '$>$') and its non-singularity. In [137, Prop. 10.44], a proof for the fact that *strictly* d.d. real symmetric matrix is positive definite is written. The proof for the case of '$\geq$' is immediate from the fact. We may consider a d.d. matrix A with non-negative diagonal entries, and for all $\varepsilon > 0$, $A + \varepsilon I$ is strictly d.d., where I is the identity matrix. Then it is concluded that $\boldsymbol{x}^T A \boldsymbol{x} \geq \sup_{\varepsilon > 0}(-\varepsilon \|\boldsymbol{x}\|^2) = 0$ for all $\boldsymbol{x} \in \mathbb{R}^n \setminus \{\mathbf{0}\}$.

is negative when $f'(x) < -f''(g(x))g'(x)^2/g''(x)$.

In case of D_{KL}, we may easily find that $D_{\mathsf{KL}}(z|x+y)$ is convex but $D_{\mathsf{KL}}(z^2|x^2+y^2)$ is not despite the convexity of D_{KL} and x^2. In fact,

$$\frac{\partial^2}{\partial x^2} D_{\mathsf{KL}}(z^2|x^2+y^2) < 0, \text{ when } (x^2+y^2)^2 + z^2(y^2-x^2) < 0. \tag{B.9}$$

Hence the original HPSS is not convex programming, though we may say that the objective function is quasiconvex with respect to each variable. Therefore, we cannot deny the existence of stationary points, on which the updating halts even though they are not the optimal.

Constrained Convex Programming and Non-Convexity of HPSS 1-A, Convexity of HPSS 1-B

A constrained optimization problem is called convex when following two conditions hold,

1. Objective function is convex.
2. Given inequality constraints $g_i(x) \leq 0$, and all the functions $g_i(x)$ are convex.

Note it is easily confirmed that a optimization problem with equality constraints $h_i(x) = 0$ is convex, if and only if all the functions h_i are affine (i.e., first-order polynomial), because both $h_i(x)$ and $-h_i(x)$ should be convex. Hence, even though S is convex, Problem 1-A is not a convex programming, because the equality constraint ${H_{n,k}}^{2\gamma} + {P_{n,k}}^{2\gamma} - {Y_{n,k}}^{2\gamma} = 0$ is not affine in the variables ${H_{n,k}}^{\gamma}$ and ${P_{n,k}}^{\gamma}$. To the contrary, Problem 1-B is a convex programming, because the equality constraints are affine, and the objective function S is convex.

B.1.2 Initial Values as a Heuristics

Thus the problems are not well-posed optimization problems. Therefore, just decreasing the objective function does not sufficiently result in a good separation result in general. Indeed, in a bad condition, $\boldsymbol{H}^{\gamma}$ and $\boldsymbol{P}^{\gamma}$ are not updated at all. A trivial example of a bad initial value is $\boldsymbol{H}^{\gamma} = \boldsymbol{Y}^{\gamma}, \boldsymbol{P}^{\gamma} = 0$, in which they are not updated at all by (6.25) and (6.26).

We need additional heuristic countermeasures to find the global optimal, or at least a meaningful local optimal. One of the possible heuristic countermeasures is the use of several initial values. That is, one may start with several initial values, and update all of them concurrently, then lastly compare the values of the objective functions to pick the best of them up as a solution. We may expect that more initial values we try, more possibly we may find the global optimal or a meaningful local optimal.

However, this concurrent computation makes the algorithms more costly. This clashes with our initial thrust of constructing a simplified technique. Instead of that, we considered to start with a single initial value, which is empirically verified to work effectively. In our experiences, we have observed that the initial values i.e.,

$$\boldsymbol{H}^{\gamma} \leftarrow \boldsymbol{Y}^{\gamma}/\sqrt{2}, \quad \boldsymbol{P}^{\gamma} \leftarrow \boldsymbol{Y}^{\gamma}/\sqrt{2}, \tag{B.10}$$

result in fair separation results in many cases.

B.2 Homogeneity of Objective Functions

We must assume a proportionality of the separated spectrograms $H_{n,k}$ and $P_{n,k}$ to the input spectrogram $Y_{n,k}$. Let us consider a music signal $y(t)$ and its r times signal $ry(t)$, and apply a separation algorithm to the signal. In such a case, it should be natural to request that it must separate $ry(t)$ into $rh(t)$ and $rp(t)$, if it separates $y(t)$ into $h(t)$ and $p(t)$. On the spectrogram domain, the requirement is written as follows,

$$\forall r \geq 0,\ \boldsymbol{Y} \overset{\text{HPSS}}{\to} (\boldsymbol{H}, \boldsymbol{P}) \ \Rightarrow\ r\boldsymbol{Y} \overset{\text{HPSS}}{\to} (r\boldsymbol{H}, r\boldsymbol{P}). \tag{B.11}$$

This property is equivalent to the following property that an objective function $f(\boldsymbol{H}, \boldsymbol{P}, \boldsymbol{Y})$, where f represents S and U, should have following property for arbitrary positive real number r,

$$\underset{(\boldsymbol{H},\boldsymbol{P})}{\operatorname{argmin}} f(\boldsymbol{H}, \boldsymbol{P}, \boldsymbol{Y}) = \underset{(\boldsymbol{H},\boldsymbol{P})}{\operatorname{argmin}} f(r\boldsymbol{H}, r\boldsymbol{P}, r\boldsymbol{Y}). \tag{B.12}$$

The smoothness function $S(\boldsymbol{H}^{\gamma}, \boldsymbol{P}^{\gamma}; w)$ satisfies the condition (B.12), because

$$S((r\boldsymbol{H})^{\gamma}, (r\boldsymbol{P})^{\gamma}, w) = r^{2\gamma} S(\boldsymbol{H}^{\gamma}, \boldsymbol{P}^{\gamma}, w). \tag{B.13}$$

In addition to S, KL divergence $D_{\mathsf{KL}}(\boldsymbol{H} + \boldsymbol{P} \| \boldsymbol{W})$ added to S in Problem 2 must also satisfy the same property with the same proportionality factor, because the objective function $U = S + \mu D_{\mathsf{KL}}$ should satisfy (B.12). That is, the divergence D_{KL} should satisfy

$$D_{\mathsf{KL}}((r\boldsymbol{Y})^{\xi} \| (r\boldsymbol{H})^{\xi} + (r\boldsymbol{P})^{\xi}) = r^{2\gamma} D_{\mathsf{KL}}(\boldsymbol{Y}^{\xi} \| \boldsymbol{H}^{\xi} + \boldsymbol{P}^{\xi}).$$

Hence, $\xi = 2\gamma$.

B.3 Related Work on Harmonic/Percussive Separation

Owing to its potential usefulness, many attempts have been made to develop such methods that separate music signal into harmonic components and percussive components in the literature of music signal processing. In order to separate these components, we naturally need to utilize some properties of percussive and harmonic instruments, or, at least some features that is useful to discriminate these instruments (such as a higher order statistics in [150], etc.). Indeed, there have been plenty of signal features utilized in the series of the studies.

This section reviews the features of harmonic and percussive instruments the state-of-the-art methods focused on, as well as the way in which these features are utilized in the methods. We largely classified the approaches of utilizing the features into following three classes, and review them orderly.

1. Detection of percussive (or harmonic, or both) sounds followed by a signal synthesis based on the detected information.
2. Signal fragmentation using a well-known signal decomposition technique, followed by a classification of the fragments.
3. Development of congregative source separation algorithms, in which some harmonic/percussive discrimination mechanisms are incorporated. (The proposed method is classified in this class)

Detection and synthesis

This problem is called "harmonic/inharmonic separation," "drum extraction in music," etc, and many studies have been carried out, because of its potential usefulness for MIR and entertainment. including "detection and synthesis" approaches [4], [24], [28], [45], [127], [168], [167], [171], etc., that first detect specific instruments, typically percussions, and then synthesize a signal based on the detected information.

Fragmentation and Classification

Another approach is the fragmentation and classification, which firstly separates a spectrogram into many fragments, then unites some of these fragments using the estimated labels by a classification algorithm such as the support vector machine (SVM). One of the earliest work that took this approach was the work by Uhle et al. [150]. They used independent subspace analysis (ISA) in order to decompose a signal. After decomposition, it picks up drum components by a simple decision rule.

Some other works utilized non-negative matrix factorization (NMF) [90], [138] in a decomposition stage. Helen and Virtanen [57] applied NMF to this problem, in which NMF was followed by SVM. They examined 15 features in all, including 8 spectral features such as *MFCC* and 7 temporal features such as *Periodicity* [56], etc., and verified that some of these features are especially useful in the discrimination. There are still other studies which addressed the relevant approaches, e.g., [77], [78], [104], [116], [136], [166]. A relevant survey on audio-based music features are presented in [41], in which plenty of audio features are listed that are used in music recognition and annotation.

Congregative Signal Decomposition Technique

The third approach considers development of a novel signal decomposition technique in which some harmonic/percussive discrimination mechanisms are built in. That is, the third approach considers formulations of algorithms that separate and classify music signals simultaneously, by considering the properties of harmonic and percussive sounds. The proposal method is most related to this approach.

Specifically, some of these methods were formulated as "a constrained NMF," which has another cost function that models the properties of the instruments, in addition to the basic cost function of the standard NMF. This approach is technically challenging, and many methods have been proposed recently.

It has been pointed out that NMF is well compatible with probabilistic inferences framework [159]. In this perspective, these additional cost functions can be formulated as a prior distribution. On the basis of this concept, some methods were formulated in the framework of Bayesian inference, MAP estimation, etc. Note, not only the studies that explicitly mentioned the Bayesian framework such as [21], [152], but also the studies above in which the fact was not mentioned specifically may also be interpreted in this way. One of the studies which explicitly mentioned the framework is the study of Dikmen and Cemgil [21]. They modeled the prior distribution of the decomposition matrix of NMF taking into *temporal (frequency) smoothness of harmonic (percussive) spectrogram* account. Specifically, the prior distribution is modeled by gamma Markov chain, a model of a sequence of positive values [15]. These kinds of distribution are also exploited in other methods which were based on tensor factorization, proposed by FitzGerald et al. [33], [34], [35]. These assumptions on harmonic and percussive instruments mentioned in [21], [33], [159] are essentially quite similar to the assumptions of this thesis, though the way of formulation is different.

Appendix C

Key Transposition in Karoake System

C.1 Key Transposition with Fixed Spectral Envelope

C.1.1 Introduction and Related Work

Karaoke is not merely a singing suppression. The real-world karaoke systems have many functions including transposition. This section describes a key transposition technique, proposed by Mizuno [103].

In case of an ordinary MIDI-based karaoke system, It is easy to transpose pitch, because it is achieved just by adding an integer to the MIDI note numbers. In case of audio signals, it is not as easy as the case of MIDI, and many pitch conversion techniques have been proposed.

Classical techniques include wave concatenation [7], [12], PSOLA [16], [30], which effectively work for single source audio signals. A drawback of these methods is that they are not basically suitable for multi-source signals, such as a music signal. Some other pitch conversion methods includes phase vocoder [37] [86], STRAIGHT [75], and etc., which can be used for these multi-source signals. The drawback of these method, however, is that they are not necessarily suitable for real-time processing, though many efforts have been devoted e.g., [3]. In addition, some of the existing techniques have drawback that the timbre, i.e., the spectral envelope, changes in the conversion procedure, and the timbre sounds quite different, if the sounds are transposed in many intervals. (See Figure C.1 (b)).

In summary, the state-of-the-art pitch conversion techniques typically have some of following problems.

Notes on Appendix C: This chapter consists of some materials in [103], though the section C.2 was done by the author.

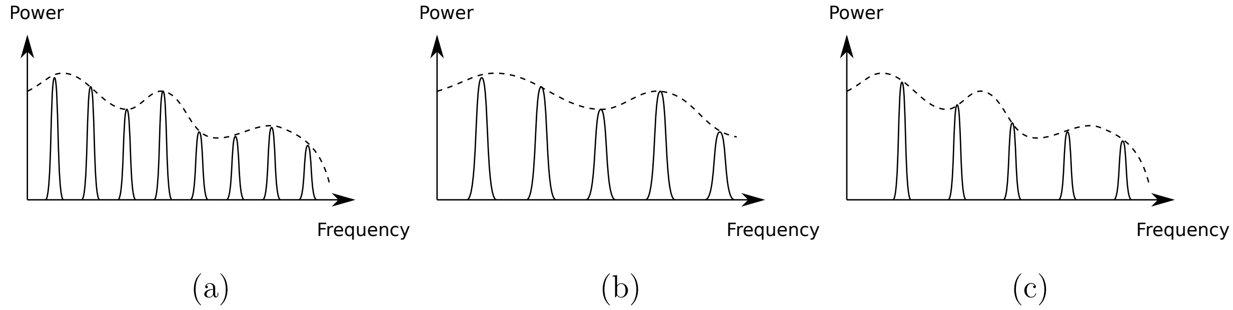


Figure C.1:: (a) Original Spectrum. Dotted contour indicates the envelope of the spectrum. (b) Straightforward expansion of the spectrum. Spectral envelope is expanded, and therefore, the timbre (spectral envelope) sounds different from the original spectrum. (c) Fixed-envelope pitch modification. Harmonic series are expanded, but spectral envelope is kept fixed.

1. Applicable only to single pitch signal (not applicable to multi-pitch signals).
2. Computationally expensive.
3. The timbre (spectral envelope) typically changes from the original signal.

On the based of these motivation, an efficient pitch modification, keeping the spectral envelope fixed, for multi-pitch signal, is proposed by Mizuno et al. [102], based on following two techniques.

Power Spectrogram Conversion Expand/shrink the given power spectrogram in the direction of frequency axis, keeping the spectral envelope fixed.

Waveform Synthesis from Power Spectrogram: Obtain the waveform from the modified power spectrogram using real-time iterative spectrogram inversion with look-ahead (RTISI-LA) [170].

C.1.2 Power Spectrogram Conversion with Fixed Spectral Envelope

A simplest pitch conversion is done by the "resampling" of spectrum. That is, by simply expanding/shrinking the short-time spectrum, then the pitch may change, as described in Figure C.1 (b). The resampling is simply done as follows. Given a vector $\boldsymbol{s} = (s_0, \ldots, s_{L-1})$, and apply DFT to it, then a frequency representation $\tilde{\boldsymbol{s}} = (\tilde{s}_0, \ldots, \tilde{s}_{L-1})$ is obtained. Then, apply inverse DFT of length L', to the obtained spectrum $\tilde{\boldsymbol{s}}$. The ratio of expansion is written as

$$\alpha = L'/L \approx 2^{n/12}, \tag{C.1}$$

where n [semitone] is the interval of key transposition. For details, see Algorithm C.2, ignoring the issues on LPC. See also Appendix C.2.

A drawback of this simplest method is that the timbre, i.e. spectral envelope, of the spectrum also changes. In order to avoid this problem, Mizuno's method first decomposes a spectrum into the envelope and the pitch by LPC analysis [69], [118, Ch.3]. Thus following data are obtained from a short-term spectrum.

- Spectral envelope (LPC coefficients)
- Pitch information (Residuals of LPC analysis)

Then the method changes the pitch by a resampling technique (Algorithm C.3)

$$\text{Up Sampling:} \quad \mathsf{DFT}^{-1}(L) \circ (\text{Zero padding}) \circ \mathsf{DFT}(L') \quad \text{if} \quad L' < L \tag{C.2}$$

$$\text{Down Sampling:} \quad \mathsf{DFT}^{-1}(L) \circ (\text{Projection}) \circ \mathsf{DFT}(L') \quad \text{if} \quad L' > L \tag{C.3}$$

while the spectral envelope is kept fixed, as described in Figure C.1 (c). In other words, pitch conversion, with fixed spectral envelope, can be achieved by changing the pitch of the residual of LPC analysis, which is followed by the multiplication of the envelope. Whole procedure is described in Algorithm C.2

C.1.3 Wave Synthesis from Modified Spectrogram

The wave segment obtained above, however, has a problem that the neighboring wave segments $s_n(t)$ and $s_{n+1}(t)$ are not consistent. That is, there should ideally exist a signal $s^*(t)$ that holds following

$$s_n(t) = w(t + c_n)s^*(t) \qquad t \in U_n \tag{C.4}$$

$$s_{n+1}(t) = w(t + c_{n+1})s^*(t) \qquad t \in U_{n+1} \tag{C.5}$$

where $w(t)$ is a window function, $c_n = n\varsigma$, ς is frame shift, and U_n is the domain of each segment

$$U_n = \{t \in \mathbb{Z} | n\varsigma - L/2 \leq t < n\varsigma + L/2\}, \tag{C.6}$$

Algorithm C.2 Key Conversion (Transposition) on Power Spectrogram Domain

```
Calculate a new frame length $L'$ on the basis of Appendix C.2.
$K = L/2 + 1$, $K' = L'/2 + 1$.
Short-term waveform $\boldsymbol{s} = (s_1, s_2, \cdots, s_{L'}) \in \mathbb{R}^{L'}$ is given.
if $L \neq L'$ then
    Apply LPC analysis of filter coefficients $p$, to the waveform $\boldsymbol{s}$.
    (Thus coefficients $\boldsymbol{c} = (c_1, \cdots, c_p)$ and residuals $\boldsymbol{r} = (r_1, \cdots, r_L)$ are obtained.)
    Apply the resampling algorithm (Algorithm C.3)
else if $L = L'$ then
    Do nothing.
end if
Thus the modified short-term waveform of length $L$ is obtained.
```

Algorithm C.3 Resampling of the Residuals of LPC analysis

```
Apply $\mathsf{DFT}(L')$ to $\boldsymbol{r}$, and obtain $\hat{\boldsymbol{r}} = (\hat{r}_1, \hat{r}_2, \cdots, \hat{r}_{K'}) \in \mathbb{C}^{K'}$.    ▷ DFT
if $L' < L$ (i.e., $K' < K$) then
    $\hat{\boldsymbol{r}}' \leftarrow (\hat{r_1}, \cdots, \hat{r_{K'}}, 0, 0, \cdots, 0) \in \mathbb{C}^K$    ▷ Zero Padding
else if $L' > L$ (i.e., $K' > K$) then
    $\hat{\boldsymbol{r}}' \leftarrow (\hat{r_1}, \cdots, \hat{r_K}) \in \mathbb{C}^K$, i.e., trim away $(\hat{r_{K+1}}, \cdots, \hat{r_{K'}})$    ▷ Projection
end if
$\boldsymbol{r}' \leftarrow \mathsf{DFT}^{-1}(L)[\hat{\boldsymbol{r}}'] \in \mathbb{R}^L$.    ▷ Inverse DFT
Thus the input signal $\boldsymbol{r} \in \mathbb{R}^{L'}$ of pitch $f$, is converted into $\boldsymbol{r'} \in \mathbb{R}^L$ of pitch $fL'/L$.
Synthesize the waveform $\boldsymbol{s}$ using the LPC coefficients $\boldsymbol{c}$ and the modified residual $\boldsymbol{r}'$.
```

Algorithm C.4 Griffin & Lim's Wave Synthesis from Power Spectrogram [52]

```
Given an amplitude spectrogram $\boldsymbol{S} = (S_{n,k}) \in \mathbb{R}^{N \times K}$,
Set random value to the initial phase spectrogram $\boldsymbol{\phi} = (\phi_{n,k}) \in [0, 2\pi)^{N \times K}$
repeat
    $\boldsymbol{C} \leftarrow \boldsymbol{S} \exp\{\sqrt{-1}\boldsymbol{\phi}\}$, where $\boldsymbol{C} \in \mathbb{C}^{N \times K}$
    $\tilde{\boldsymbol{C}} \leftarrow \mathsf{STFT}(L) \left[\mathsf{STFT}^{-1}(L)[\boldsymbol{C}]\right]$
    $\boldsymbol{\phi} \leftarrow \angle\tilde{\boldsymbol{C}}$
until the values almost converge
A wave form $s(t) = \mathsf{STFT}^{-1}(L)[\boldsymbol{S} \exp\{\sqrt{-1}\boldsymbol{\phi}\}]$ is obtained.
```

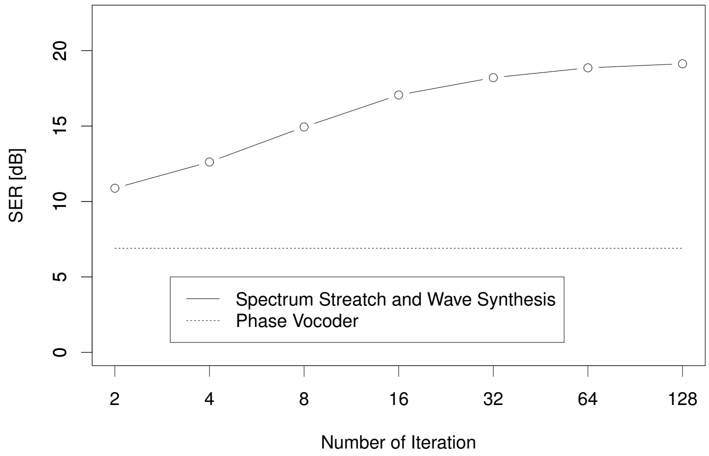


Figure C.5:: Relation between SER and the number of iteration in the proposed pitch modification The ratio of pitch was $\alpha = 1.87$. The size of sliding block was $I = 10$. The horizontal axis indicates the number of iteration (let us denote it N) in single update, i.e., in a single update procedure, the technique apply the updating formulae I times, and the actual iteration number for each bin is $I \times N$. The values of this graph are excerpted from [103, Ch. 2].

In other words, the segments do not satisfy

$$\frac{s_n(t)}{w(t+c_n)} = \frac{s_{n+1}(t)}{w(t+c_{n+1})} \text{ for all } t \in U_n \cap U_{n+1}. \tag{C.7}$$

Then, in order to connect these wave segments, we need to employ a technique to reconstruct waveform from a modified power spectrogram.

It is achieved by the techniques, such as the one proposed by Griffin and Lim [52], and its extension to real-time processing, RTISI-LA, proposed by Zhu et al. [170], and a work by Le Roux [87], [88], [89]. As observed in its name, the method of Zhu [170] works in real time. Above all we used RTISI-LA in this chapter.

The procedure of the Griffin & Lim's wave synthesis from spectrogram, which is the basic method of the RTISI-LA, is shown in Algorithm C.4. RTISI-LA is the real-time version of the algorithm, which is similar to the sliding HPSS.

C.1.4 Performance Evaluation of Pitch Transposition

We evaluated the performance of the pitch conversion by signal to error ratio (SER) [53], which is a spectral distance between two spectrograms. The performance of pitch conversion in a case of single pitch conversion based on the approach is shown in Figure C.5. This shows that the RTISI-LA based approach outperforms the quality of phase vocoder even with small number of iterations. It also shows that the larger number of iterations can improve the quality of sound.

C.2 Approximate Values of $2^{n/12}$

Frame Length for Resampling

In the resampling in key transposition, we used FFT of size $L' = \alpha L$, where $\alpha = 2^{n/12}$, and n is the interval of key transposition. For example, if we modify a signal of frequency 440 Hz into 880 Hz, $\alpha = 2 = 2^{12/12}$. This case is not indeed a problem, because FFT of size $2L$ is effective when FFT of size L is effective. However, when it comes to a case of other intervals such as "Major 3rd", i.e. $\alpha = 2^{5/12}$, the value should not be used because of the reasons below, but we must select another α carefully, instead. This chapter describes the way to decide α.

In the Equal Temperament, which is currently the most commonly used tuning, the frequency ratio that correspond to a semitone is $2^{1/12}$, and those of n semitones are $2^{n/12}$. Therefore, the modified frame length is written as $L' = 2^{n/12}L$. However, in these cases, L' is not an integer in general. Therefore, we must consider approximating L' by an integer. An easiest way to make L' integer is to just simply round it. That is,

$$L' = \lceil 2^{n/12} L \rfloor. \tag{C.8}$$

There is, however, still a problem that the value L' is not necessarily a product of small primes. For example, if $L = 1024$ and $n = 6$, then

$$L' = L \times 2^{6/12} = 1024 \times 2^{6/12} \approx 1448.1547 \approx 1448 = 2^3 \times 181. \tag{C.9}$$

where 181 is a prime, which is not sufficiently small in the literature of FFT. In general FFT algorithms, it is required that the frame length is a product of small primes. Specifically, if FFT size is expressed as $L = \prod_i p_i^{e_i}$, where p_i is prime and $e_i \in \mathbb{N}$, then the computation complexity of FFT is $O(L \sum_i e_i p_i)$. Because of the reason, FFTW3 requires that the size of DFT should be a product of small primes smaller than 17, i.e.,

$$L = 2^{e_2} 3^{e_3} 5^{e_5} 7^{e_7} 11^{e_{11}} 13^{e_{13}} 17^{e_{17}}, \quad e_2, e_3, \cdots \in \mathbb{N}. \tag{C.10}$$

In addition, it also requires $e_{13} + e_{17} \leq 1$ for efficiency [39].

Frame Lengths Composed of Small Primes

Thus we must consider seeking frame lengths of simpler integers. That is, we need to approximate the values by the simple integers. For example, let us approximate $2^{6/12}$ by $45/32 \approx 2^{5.902/12}$, and L' can be expressed as a product of small primes in this case as follows,

$$L' = 1024 \times 2^{6/12} \approx 1024 \times 45/32 = 1440 = 2^5 \times 3^2 \times 5, \tag{C.11}$$

then FFT of size $L' = 1440$ is more effective than that of 1448.

On the basis of the idea, we found the approximate values of $2^{n/12}$ for positive and negative integers $-12 \leq n \leq 12$, which can be expressed as $3^\alpha 5^\beta / 2^\gamma$, assuming that $L = 2^n$ where $n \geq 10, n \in \mathbb{N}$, in order to make the form of L' as follows,

$$L' = L \times 3^\alpha 5^\beta / 2^\gamma = 2^{\log_2 L - \gamma} \times 3^\alpha \times 5^\beta, \tag{C.12}$$
$$\text{where} \quad \gamma \leq 10 \leq \log_2 L, \quad \alpha, \beta \geq 0.$$

Table C.6 shows the list of approximate values we have found. These values makes FFT less costly than just using a rounded values of $2^{n/12}L$. Note it is also possible making the original frame length L a product of small primes. However, we did not do that, because the original FFT of size L is much more frequently executed than the FFT of size L', in the procedure of wave synthesis algorithms.

Let us compare the frequency ratios defined above to the tunings which are commonly used in music performance, in order to show that the defined frequency ratios are not critically distant from the common tunings. We compared the values to the some of standard tunings; Equal Temperament (ET), Just Intonation (JI), and Mean tone (MT). Table C.7 shows the three types of tunings and the frame length ratios defined above. It shows that the proposed ratios do not have critical errors from ET, comparing to JI and MT.

Table C.6: Approximate values of $2^{n/12}$.

positive n	negative n
$2^{1/12} \approx 135/128$	$2^{-1/12} \approx 15/16$
$2^{2/12} \approx 9/8$	$2^{-2/12} \approx 225/256$
$2^{3/12} \approx 1215/1024$	$2^{-3/12} \approx 27/32$
$2^{4/12} \approx 5/4$	$2^{-4/12} \approx 405/512$
$2^{5/12} \approx 675/512$	$2^{-5/12} \approx 3/4$
$2^{6/12} \approx 45/32$	$2^{-6/12} \approx 45/64$
$2^{7/12} \approx 3/2$	$2^{-7/12} \approx 675/1024$
$2^{8/12} \approx 405/256$	$2^{-8/12} \approx 5/8$
$2^{9/12} \approx 27/16$	$2^{-9/12} \approx 75/128$
$2^{10/12} \approx 225/128$	$2^{-10/12} \approx 9/16$
$2^{11/12} \approx 15/8$	$2^{-11/12} \approx 135/256$

Table C.7: Comparison between the products of small primes and the frequency ratios of several common tunings. Percentage indicates the distance in cent, from the Equal Temperament.

Musical	Standard Tunings									
Interval	ET	JI			MT			FFT frame length		
Unison	$2^{0/12}$	1	$= 2^{0/12}$	0%	1	$= 2^{0/12}$	0%	1	$= 2^{0/12}$	0%
minor 2nd	$2^{1/12}$	$16/15$	$\approx 2^{1.11/12}$	+11%	$5\sqrt[4]{5^3}/16$	$\approx 2^{0.76/12}$	−24%	$135/128$	$\approx 2^{0.92/12}$	−8%
Major 2nd	$2^{2/12}$	$9/8$	$\approx 2^{2.04/12}$	+4%	$\sqrt{5}/2$	$\approx 2^{1.93/12}$	−7%	$9/8$	$\approx 2^{2.04/12}$	+4%
minor 3rd	$2^{3/12}$	$6/5$	$\approx 2^{3.16/12}$	+16%	$4/\sqrt[4]{5^3}$	$\approx 2^{3.10/12}$	+10%	$1215/1024$	$\approx 2^{2.96/12}$	−4%
Major 3rd	$2^{4/12}$	$5/4$	$\approx 2^{3.86/12}$	−14%	$5/4$	$\approx 2^{3.86/12}$	−14%	$5/4$	$\approx 2^{3.86/12}$	−14%
Perfect 4th	$2^{5/12}$	$4/3$	$\approx 2^{4.98/12}$	−2%	$2/\sqrt[4]{5}$	$\approx 2^{5.03/12}$	+3%	$675/512$	$\approx 2^{4.78/12}$	−22%
Tritone	$2^{6/12}$	$7/5$	$\approx 2^{5.83/12}$	−17%	$5\sqrt{5}/8$	$\approx 2^{5.79/12}$	−21%	$45/32$	$\approx 2^{5.90/12}$	−10%
Perfect 5th	$2^{7/12}$	$3/2$	$\approx 2^{7.02/12}$	+2%	$\sqrt[4]{5}$	$\approx 2^{6.97/12}$	−3%	$3/2$	$\approx 2^{7.02/12}$	+2%
minor 6th	$2^{8/12}$	$8/5$	$\approx 2^{8.14/12}$	+14%	$25/16$	$\approx 2^{7.73/12}$	−27%	$405/256$	$\approx 2^{7.94/12}$	−6%
Major 6th	$2^{9/12}$	$5/3$	$\approx 2^{8.84/12}$	−16%	$\sqrt[4]{5^3}/2$	$\approx 2^{8.90/12}$	−10%	$27/16$	$\approx 2^{9.06/12}$	+6%
minor 7th	$2^{10/12}$	$16/9$	$\approx 2^{9.96/12}$	−4%	$4/\sqrt{5}$	$\approx 2^{10.1/12}$	+1%	$225/128$	$\approx 2^{9.77/12}$	−23%
Major 7th	$2^{11/12}$	$15/8$	$\approx 2^{10.9/12}$	+9%	$5\sqrt[4]{5}/4$	$\approx 2^{10.8/12}$	+8%	$15/8$	$\approx 2^{10.9/12}$	+9%
Octave	$2^{12/12}$	2	$= 2^{12/12}$	0%	2	$= 2^{12/12}$	0%	2	$= 2^{12/12}$	0%

Appendix D

Quick Instruction on Some Computations in This Thesis

D.1 STFT and Inverse STFT

We may try following functions to execute STFT in GNU R.

```
ffthalf <- function(x) fft(x)[1:(length(x)/2+1)]
win.sine <- function(x) sin(x/length(x) * pi)

offset <- function(fl, shi)(fl/2 + ((fl/2) %/% shi) * shi)

stft <- function(signal, fl = 512, shi = 256, win = win.sine){
  n <- (length(signal) + offset(fl,shi) * 2) %/% shi
  z <- numeric( offset(fl,shi) )
  signal <- c(z, signal, z)
  return( do.call("rbind", lapply( 1:n,
    function(x)( ffthalf( win(1:fl) * signal[1:fl + shi * (x - 1)])))))
}
```

Then we may try the following commands, and a spectrogram will be displayed.

```
# continued
fs <- 8000
fl <- 512
shi <- 256
sw <- function(freq) sin(1:(16000 * 10) / fs * freq * 2 * pi)

signal <- sw(440) + sw(880)/2 + sw(1320)/3 + sw(1760)/4 + sw(2200) / 5
spectrogram <- stft(signal, fl, shi)

image(((0:nrow(spectrogram)-1)-(fl/2) %/% shi- 1) * shi / fs,
       0:(ncol(spectrogram)-1) / fl * fs,
        abs(spectrogram)**0.6, xlab="Time␣[s]", ylab = "Frequency␣[Hz]")
```

An example of inverse STFT is as follows,

```
# continued
iffthalf <- function(x)
  Re(fft(c(x,Conj(rev(x[2:(length(x)-1)]))),inverse=T))/(2*length(x)-2)

win.norm <- function(fl, shi, win.fwd = win.sine, win.inv = win.sine){
  z <- numeric(fl * 2);
  tmp <- c( z, win.fwd(1:fl) * win.inv(1:fl), z)
  colSums(do.call("rbind", lapply( -(fl %/% shi + 2):(fl %/% shi + 2),
    function(x)( tmp[(fl * 2) + shi * x + 1:fl]))))}

istft <- function(spec, shi = 256, win.fwd = win.sine, win.inv = win.sine,
    total.len){
  fl <- ncol(spec) * 2 - 2
  n <- nrow(spec)
  win.modify <- win.norm(fl, shi, win.fwd, win.inv)
  sigs <- do.call("rbind", lapply( 1:n, function(x)
    (iffthalf(spec[x,]) * win.inv(1:fl) / win.modify[1:fl])))

  sig <- numeric(total.len + fl * 5) * 0
  for(x in 1:n)
    sig[shi * (x - 1) + 1:fl] <- sig[shi * (x - 1) + 1:fl] + sigs[x,1:fl]

  return(sig[offset(fl,shi) + 1:total.len])
}
```

Indeed, we may try the following command, and find that the difference between the original signal $x(t)$ and $\mathsf{STFT}^{-1}(L)[\mathsf{STFT}(L)\,[x(t)]]$ is quite small, which should be regarded as numerical errors.

```
# continued
spectrogram <- stft(signal, shi = shi)
signal2 <- istft(spectrogram, shi = shi total.len=length(signal))
plot(signal - signal2, type = "l")
```

D.2 HPSS

Using these functions we may easily implement HPSS in GNU R. An example is the following code.

```
time.averaging <- function(H, N){
  n.frame <- nrow(H)
  return (do.call("rbind",lapply(
    1:n.frame, # for all frames
    function(x){
      range <- (max(1, x - N) : min(n.frame - 1, x + N))
      return (colMeans(H[range,])) }))))}

freq.averaging <- function(P, K){
  return ( t(time.averaging( t(P), K)))
}

rowMedian <- function(x)(apply(x,1,median))
colMedian <- function(x)(apply(x,2,median))

# This is not strictly identical to the algorithm described in this
   dissertation.
# The updating formulae are applied for each bin ORDERLY,
# but this program update all the bins SIMULTANEOUSLY.
# Nevertheless the difference between them is not very large empirically.

HPSS.new <- function(H, P, Y, N, K){

  b1 <- time.averaging(H, N)
  b2 <- freq.averaging(P, K)

  R <- sqrt(b1 ** 2 + b2 ** 2)
  H <- b1 / R * Y
  P <- b2 / R * Y

  return (list(H=H,P=P))
}
```

We may execute HPSS using the following code,

```
library(tuneR)
input <- readWave("input_filename.wav")@left

fl <- 512;
shi <- fl/4;
win <- win.sine

y.spec <- stft(input , fl, shi, win)

H <- abs(y.spec) ** 0.5 / sqrt(2)
P <- abs(y.spec) ** 0.5 / sqrt(2)
```

```
Y <- abs(y.spec) ** 0.5

for(i in 1:30){
  tmp <- HPSS.new(H, P, Y, N=10, K=10)
  H <- tmp$H
  P <- tmp$P
}

image(H)
image(P)
```

In order to enable `tuneR` package, you may try the following command.

```
install.packages("tuneR")
```

For more efficient implementation, more efficient languages should be used such as C++, etc. The source codes written in C++ are available at the author's Github (Feb. 2014).

Bibliography

[1] S. A. Abdallah and M. D. Plumbley, “Unsupervised analysis of polyphonic music by sparse coding,” *IEEE Transactions on Neural Networks*, vol. 17, no. 1, pp. 179–196, 2006. ▷ page 2.

[2] L. Atlas and C. Janssen, “Coherent modulation spectral filtering for single-channel music source separation,” in *Proceedings of 2005 IEEE International Conference on Audio, Speech, and Signal Processing* (*ICASSP 2005*), 2005, pp. 461–464. ▷ page 20.

[3] H. Banno, H. Hata, M. Morise, T. Takahashi, T. Irino, and H. Kawahara, “Implementation of realtime STRAIGHT speech manipulation system: Report on its first implementation,” *Acoustical Science and Technology*, vol. 28, no. 3, pp. 140–146, 2007. ▷ page 120.

[4] D. Barry, D. Fitzgerald, and E. Coyle, “Drum source separation using percussive feature detection and spectral modulation,” in *Proceedings of IEE Irish Signals and Systems Conference*, 2005. ▷ page 117.

[5] R. Bencina and P. Burk, “PortAudio – an open source cross platform audio API,” in *Proceedings of International Computer Music Conference 2001* (*ICMC 2001*), 2001, pp. 263–266. ▷ pages 60 and 62.

[6] C. M. Bishop, *Pattern Recognition and Machine Learning.* Springer, 2007. ▷ page 46.

[7] A. W. Black and N. Campbell, “Optimising selection of units from speech databases for concatenative synthesis,” 1995, pp. 581–584. ▷ page 120.

[8] S. Boll, “Suppression of acoustic noise in speech using spectral subtraction,” *IEEE Transactions on Audio, Speech, and Signal Processing*, vol. 27, no. 2, pp. 113–120, 1979. ▷ page 4.

[9] K. Briggs, “Diagonally dominant matrix,” from MathWorld–A Wolfram Web Resource, created by Eric W. Weisstein. [Online, viewed on Sep. 2013] http://mathworld.wolfram.com/DiagonallyDominantMatrix.html. ▷ page 114.

[10] J. C. Brown, “Calculation of a constant Q spectral transform,” *Journal of Acoustic Society of America*, vol. 89, no. 1, pp. 425–434, 1991. ▷ page 112.

[11] ——, “An efficient algorithm for the calculation of a constant Q transform,” *Journal of Acoustic Society of America*, vol. 92, no. 5, pp. 2698–2701, 1992. ▷ page 112.

[12] N. Campbell, “CHATR: A high-definition speech re-sequencing system,” in *Proceedings of ASA/ASJ Joint Meeting*, 1996, pp. 1223–1228. ▷ page 120.

[13] C. Cao and M. Li, “Singing melody extraction in polyphonic music by harmonic tracking,” *Extended Abstract of MIREX*, 2009. ▷ page 50.

[14] C. Cao, M. Li, J. Liu, and Y. Yan, “Singing melody extraction in polyphonic music by harmonic tracking,” *Proceedings of Internationl Conference on Music Information Retrieval* (*ISMIR 2007*), 2007. ▷ pages 2 and 40.

[15] A. T. Cemgil and O. Dikmen, “Conjugate gamma markov random field for modelling nonstationary sources,” in *ICA*, 2007, pp. 697–705. ▷ page 118.

[16] F. J. Charpentier and M. G. Stella, “Diphones synthesis using an overlap-add technique for speech waveforms concatenation,” in *Proceedings of 1986 IEEE International Conference on Audio, Speech, and Signal Processing* (*ICASSP 1986*), 1986, pp. 2015–2018. ▷ page 120.

[17] A. Cichocki, R. Zdunek, and S. Amari, “Csiszár’s divergences for non-negative matrix factorization: Family of new algorithms,” in *Artificial Intelligence and Soft Computing – ICAISC 2006 Extended Smart Algorithms for Non-Negative Matrix Facotorization*, 2006, p. 548. ▷ page 86.

[18] L. Cohen, *Time-Frequency Analysis.* Prentice Hall, 1995. ▷ pages 12 and 111.

[19] I. Daubechies, *Ten Lectures on Wavelets.* Society for Industrial and Applied Mathematics, 1992. ▷ pages 11 and 111.

[20] *Digital Content White Paper 2008* (デジタルコンテンツ白書 2008). Digital Contents Association of Japan (財団法人デジタルコンテンツ協会) Editorial: Commerce and Information Policy Bureau, Ministry of Economy, Trade, and Industry, Japan (経済産業省 商務情報制作局 監修), 2008, (in Japanese). ▷ page 3.

[21] O. Dikmen and T. Cemgil, “Unsupervised single-channel source separation using bayesian NMF,” in *Proceedings of 2009 IEEE Workshop on Applications of Signal Processing to Audio and Acoustics* (*WASPAA*), 2009, pp. 93–96. ▷ pages 79, 118, and 119.

[22] J. S. Downie, “The music information retrieval evaluation exchange (2005–2007): A window into music information retrieval research,” *Acoustical Science and Technology*, vol. 29, no. 4, pp. 247–255, 2008. ▷ page 40.

[23] K. Dressler, “Audio melody extraction for mirex 2009,” *Extended Abstract of MIREX*, 2009. ▷ page 50.

[24] Z. Duan, Y. Zhang, C. Zhang, and Z. Shi, “Unsupervised single-channel music source separation by average harmonic structure modeling,” *IEEE Transactions on Audio, Speech, and Signal Processing*, vol. 16, no. 4, pp. 766–778, 2008. ▷ page 117.

[25] J. L. Durrieu, G. Richard, and B. David, “Singer melody extraction in polyphonic signals using source separation methods,” *Proceedings of 2008 IEEE International Conference on Audio, Speech, and Signal Processing* (*ICASSP 2008*), pp. 169–172, 2008. ▷ pages 2 and 40.

[26] J.-L. Durrieu, G. Richard, and B. David, “A source/filter approach to audio melody extraction,” *Extended Abstract of MIREX*, 2009. ▷ page 50.

[27] J.-L. Durrieu, G. Richard, B. David, and C. Fevotte, “Source/filter model for unsupervised main melody extraction from polyphonic audio signals,” *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 18, no. 3, pp. 564–575, 2010. ▷ pages 2 and 40.

[28] C. Duxbury, M. Davies, and M. Sandler, “Separation of transient information in music audio using multiresolution analysis techniques,” in *Proceedings of Digital Audio Effects 2001* (*DAFx*), 2001. ▷ pages 79 and 117.

[29] K. El-Maleh, M. Klekn, G. Petrucci, and P. Kabal, “Speech/music discrimination for multimedia applications,” *Proceedings of 2002 IEEE International Conference on Audio, Speech, and Signal Processing* (*ICASSP 2002*), pp. 2445–2448, 2002. ▷ page 74.

[30] E.Moulines and F. Charpentier, “Pitch-synchronous waveform processing techniques for text-to-speech synthesis using diphones,” *Speech Communication*, vol. 9, no. 5-6, pp. 453–467, 1990. ▷ page 120.

[31] Y. Ephraim and D. Malah, “Speech Enhancement Using A Minimum-Mean Square Error Short-Time Spectral Amplitude Estimator,” *IEEE Transactions on Audio, Speech, and Signal Processing*, vol. 32, no. 6, pp. 1109–1121, 1984. ▷ page 4.

[32] D. FitzGerald, “Harmonic/percussive separation using median filtering,” in *Proceedings of Digital Audio Effects 2010 (DAFx)*, 2010. ▷ pages xvi, xviii, 88, 95, 96, and 99.

[33] D. FitzGerald, E. Coyle, and M. Cranitch, “Using tensor factorisation models to separate drums from polyphonic music,” in *Proceedings of Digital Audio Effects 2009 (DAFx)*, 2009. ▷ pages 79 and 119.

[34] D. FitzGerald, M. Cranitch, and E. Coyle, “Extended nonnegative tensor factorisation models for musical sound source separation,” *Computational Intelligence And Neuroscience*, 2008. ▷ page 119.

[35] ——, “Musical source separation using generalised non-negative tensor factorisation models,” in *Workshop On Music And Machine Learning, International Conference On Machine Learning*, 2008. ▷ page 119.

[36] D. FitzGerald and M. Gainza, “Single channel vocal separation using median filtering and factorisation techniques,” *ISAST Transactions on Electronic and Signal Processing*, vol. 4, no. 1, pp. 62–73, 2010. ▷ page 14.

[37] J. L. Flanagan and R. M. Golden, “Phase Vocoder,” *Bell System Technical Journal*, vol. 45, pp. 1493–1509, 1966. ▷ page 120.

[38] Y. Freund and R. E. Schapire, “A decision-theoretic generailization of on-line learning and an application to boosting,” *Journal of Computing System and Science*, vol. 55, no. 1, pp. 119–139, 1997. ▷ page 75.

[39] M. Frigo and S. G. Johanson, “FFTW users manual,” [Online (Visited Sep., 2013)] http://www.fftw.org/doc/ [Online (Visited Sep., 2013)] http://www.fftw.org/fftw3.pdf. ▷ page 125.

[40] ——, “The design and implementation of FFTW3,” in *Proceedings of IEEE*, vol. 93, no. 2, 2005, pp. 216 – 231. ▷ page 98.

[41] Z. Fu, G. Fu, K. M. Ting, and D. Zhang, “A survey of audio-based music classification and annotation,” *IEEE Transactions on Multimedia*, vol. 13, no. 2, pp. 303–319, 2011. ▷ page 118.

[42] H. Fujihara, T. Kitahara, M. Goto, K. Komatani, and T. Ogata, “Singer identification based on accompaniment sound reduction and reliable frame selection,” in *Proceedings of Internationl Conference on Music Information Retrieval (ISMIR 2007)*, 2005. ▷ pages 2, 6, and 14.

[43] H. Fujihara, T. Kitahara, M. Goto, K. Komatani, T. Ogata, and H. G. Okuno, “F_0 estimation method for singing voice in polyphonic audio signal based on statistical vocal model and viterbi search,” 2006, pp. 253–256. ▷ pages 2 and 40.

[44] S. Fukayama, “Automatic music composition from japanese lyrics with probabilistic formulation,” Ph.D. dissertation, The University of Tokyo, 2013. ▷ page 5.

[45] O. Gillet and G. Richard, “Transcription and separation of drum signals from polyphonic music,” *IEEE Transactions on Audio, Speech, and Signal Processing*, vol. 16, no. 3, pp. 529–540, 2008. ▷ page 117.

[46] M. Goto, “Development of the RWC music database,” *Proceedings of the 18th Intenational Congress on Acoustics 2004 (ICA 2004)*, pp. I–553–556, 2004. ▷ pages xiv, xvi, 32, 33, 71, 81, and 82.

[47] ——, "A real-time music-scene-description system: Predominant-f0 estimation for detecting melody and bass lines in real-world audio signals," *Speech Communication*, vol. 43, pp. 311–329, 2004. ▷ pages 2 and 40.

[48] M. Goto and T. Goto, "Musicream: Integrated music-listening interface for active, flexible, and unexpected encounters with musical pieces," *IPSJ Journal*, vol. 50, no. 12, pp. 2923–2936, 2009. ▷ page 3.

[49] M. Goto, K. Yoshii, H. Fujihara, M. Mauch, and T. Nakano, "Songle: A web-service for active music listening improved by user contributions," in *Proceedings of Internationl Conference on Music Information Retrieval* (*ISMIR 2011*), 2011, pp. 311–316. ▷ page 3.

[50] M. Goto, "Active music listening interfaces based on signal processing," in *Proceedings of 2007 IEEE International Conference on Audio, Speech, and Signal Processing* (*ICASSP 2007*), 2007, pp. 1444–1447. ▷ page 3.

[51] R. Gribonval, L. Beneroya, E. Vincent, and C. Févotte, "Proposals for performance measurement in source separation," in *Proc. ICA BSS*, 2003, pp. 763–768. ▷ pages 15 and 34.

[52] D. W. Griffin and J. S. Lim, "Signal estimation from modified short-time fourier transform," *IEEE Transactions on Audio, Speech, and Signal Processing*, vol. 32, no. 2, pp. 236–243, 1984. ▷ pages 123 and 124.

[53] ——, "Speech synthesis from short-time fourier transform magnitude and its application to speech processing," in *Proceedings of 1984 IEEE International Conference on Audio, Speech, and Signal Processing* (*ICASSP 1984*), vol. 9, 1984, pp. 61–64. ▷ page 124.

[54] M. Hamasaki, H. Takeda, T. Hope, and T. Nishimura, "Network analysis of an emergent massively collaborative creation community – how can people create videos collaboratively without collaboration?" in *Proceedings of The 3rd International ICWSM Conference, AAAI*, 2009, pp. 222–225. ▷ page 54.

[55] M. Hamasaki, H. Takeda, and T. Nishimura, "Network analysis of massively collaborative creation of multimedia contents – case study of Hatsune Miku videos on nico nico douga," in *Proceedings of The 1st International Conference on Designing Interactive User Experiences for TV and Video (UXTV), ACM*, 2008, pp. 165–168. ▷ pages 3 and 54.

[56] T. Heittola and A. Klapuri, "Locating segments with drums in music signals," in *Proceedings of 3rd Internationl Conference on Music Information Retrieval* (*ISMIR 2002*), 2002. ▷ page 118.

[57] M. Helen and T. Virtanen, "Separation of drums from polyphonic music using non-negative matrix factorization and support vector machine," in *Proceedings of European Signal Processing Conference 2005* (*EUSIPCO 2005*), 2005. ▷ page 118.

[58] H. Hermansky and P. Fousek, "Multi-resolution RASTA filtering for TANDEM-basaed ASR," in *Proceedings of Interspeech 2005* (*Interspeech 2005*), 2005. ▷ pages 20 and 66.

[59] H. Hermansky and N. Morgan, "RASTA processing of speech," *IEEE Transactions on Speech and Audio Processing*, vol. 2, no. 4, pp. 578–589, 1994. ▷ pages 20 and 66.

[60] Y. Horii, "Acoustic analysis of vocal vibrato: A theoretical interpretation of data," *J. Voice*, vol. 3, pp. 36–43, 1989. ▷ pages 6 and 14.

[61] A. Horn and C. R. Johnson, *Matrix Analysis.* Cambridge University Press, 1985. ▷ page 114.

[62] C.-L. Hsu and J.-S. R. Jang, "On the improvement of singing voice separation for monaural recordings using the MIR-1K dataset," *IEEE Transactions on Audio, Speech, and Language Processing*, 2010. ▷ pages xiv, 16, 34, 35, 36, and 74.

[63] ——, "Singing pitch extraction at mirex 2010," *Extended Abstract of MIREX*, 2010. ▷ page 50.

[64] C.-L. Hsu, J.-S. R. Jang, and L.-Y. Chen, "Singing pitch extraction at mirex 2009," *Extended Abstract of MIREX*, 2009. ▷ page 50.

[65] C.-L. Hsu, D. L. Wang, J.-S. R. Jang, and K. Hu, "A tandem algorithm for singing pitch extraction and voice separation from music accompaniment," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 20, no. 5, pp. 1482–1491, 2012. ▷ pages 2, 14, and 38.

[66] C.-L. Hsu, D. L. Wann, and J.-S. R. Jang, "A trend estimation algorithm for singing pitch detection in musical recordings," in *Proceedings of 2011 IEEE International Conference on Audio, Speech, and Signal Processing* (*ICASSP 2011*), 2011, pp. 393–396. ▷ pages 2 and 38.

[67] P.-S. Huang, S. D. Chen, P. Smaragdis, and M. H.-Johnson, "Singing-voice separation from monaural recordings using robust principal component analysis," in *Proceedings of 2012 IEEE International Conference on Audio, Speech, and Signal Processing* (*ICASSP 2012*), 2012, pp. 57–60. ▷ pages 17 and 34.

[68] D. R. Hunter and K. Lange, "Quantile regression via an MM algorithm," *Journal Of Computational And Graphical Statistics*, vol. 9, pp. 60–77, 2000. ▷ page 90.

[69] F. Itakura and S. Saito, "Digital Filtering Techniques For Speech Analysis And Synthesis," *Proceedings Of The International Conference On Independent Component Analysis And Blind Signal Separation*, vol. 25-C-1, pp. 261–264, 1971. ▷ page 122.

[70] K. Itou, M. Yamamoto, K. Takeda, T. Takezawa, T. Matsuoka, T. Kobayashi, K. Shikano, and S. Itahashi, "JNAS: Japanese speech corpus for large vocabulary continuous speech recognition research," *Journal of Acoustic Society of Japan*, vol. 20, no. 3, pp. 199–206, 1999. ▷ page 71.

[71] S. Joo, S. Jo, and C. D. Yoo, "Melody extraction from polyphonic audio signal," *Extended Abstract of MIREX*, 2009. ▷ page 50.

[72] ——, "Melody extraction from polyphonic audio signal mirex 2010," *Extended Abstract of MIREX*, 2010. ▷ page 50.

[73] N. Kanedera, T. Arai, H. Hermansky, and M. Pavel, "On the relative importance of various components of the modulation spectrum for automatic speech recognition," *Speech Communication*, vol. 28, pp. 43–55, 1999. ▷ pages 20 and 66.

[74] N. Kanedera, H. Hermansky, and T. Arai, "On properties of modulation spectrum for robust automatic speech recognition," in *Proceedings of 1998 IEEE International Conference on Audio, Speech, and Signal Processing* (*ICASSP 1998*), 1998, pp. 613–616. ▷ pages 20 and 66.

[75] H. Kawahara, I. Masuda-Katsuse, and A. de Cheveigné, "Restructuring speech representations using a pitch-adaptive time-frequency smoothing and an instantaneous-frequency-based f0 extraction," *Speech Communication*, vol. 27, pp. 187–207, 1999. ▷ page 120.

[76] H. Kenmochi and H. Ohshita, "Vocaloid – commercial singing synthesizer based on sample concatenation," *Proceedings of Interspeech 2007*, pp. 4010–4011, 2007. ▷ pages 3 and 5.

[77] M. Kim, J. Yoo, K. Kang, and S. Choi, "Blind rhythmic source separation: Nonnegativity and repeatability," in *Proceedings of 2010 IEEE International Conference on Audio, Speech, and Signal Processing* (*ICASSP 2010*), 2010. ▹ page 118.

[78] ——, "Nonnegative matrix partial co-factorization for spectral and temporal drum source separation," *IEEE Journal of Selected Topics in Signal Processing*, vol. 5, no. 6, p. 1192, 2011. ▹ page 118.

[79] Y. E. Kim and B. Whitman, "Singer identification in popular music recordings using voice coding features," in *Proceedings of 3rd Internationl Conference on Music Information Retrieval* (*ISMIR 2002*), 2002, pp. 164–169. ▹ page 6.

[80] B. E. D. Kingsbury, N. Morgan, and S. Greenberg, "Robust speech recognition using the modulation spectrogram," *Speech Communication*, vol. 25, 1998. ▹ pages 20 and 66.

[81] A. Klapuri, "Multiple fundamental frequency estimation based on harmonicity and spectral smoothness," *IEEE Transactions on Speech and Audio Processing*, vol. 11, no. 6, pp. 804–816, 2003. ▹ pages 2 and 40.

[82] ——, "Multipitch analysis of polyphonic music and speech signals using an auditory model," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 16, no. 2, pp. 244–266, 2008. ▹ pages 2 and 40.

[83] A. Klapuri and M. Davy, *Signal Processing Methods for Music Transcription.* Springer, 2006. ▹ pages 2, 11, 12, and 108.

[84] H. Lachambre, R. André-Obrecht, and J. Pinquier, "Singing voice detection in monophonic and polyphonic contexts," in *Proceedings of European Signal Processing Conference 2009* (*EUSIPCO 2009*), 2009, pp. 1344–1348. ▹ page 20.

[85] M. Lagrange, L. G. Martins, J. Murdoch, and G. Tzanetakis, "Normalized cuts for predominant melodic source separation," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 16, no. 2, pp. 278–290, 2008. ▹ page 16.

[86] J. Laroche and M. Dolson, "Improved phase vocoder time-scale modification of audio," *IEEE Transactions on Speech and Audio Processing*, vol. 7, no. 3, pp. 323–332, 1999. ▹ page 120.

[87] J. Le Roux, "Exploiting regularities in natural acoustical scenes for monaural audio signal estimation, decomposition, restoration and modification," Ph.D. dissertation, The University of Tokyo, 2009. ▹ page 124.

[88] J. Le Roux, N. Ono, and S. Sagayama, "Explicit consistency constraints for stft spectrograms and their application to phase reconstruction," in *Proceedings of Workshops on Statistical and Perceptual Audition* (*SAPA*), 2008. ▹ page 124.

[89] J. Le Roux, E. Vincent, Y. Mizuno, H. Kameoka, N. Ono, and S. Sagayama, "Consistent wiener filtering: スペクトログラム無矛盾性を保証する一般化時間周波数マスクの設計法," in *Proceedings of Acoustic Society of Japan Autumn Meeting* (日本音響学会講演論文集 (秋)), 2010, (in Japanese). ▹ page 124.

[90] D. D. Lee and H. S. Seung, "Algorithms for non-negative matrix factorization," in *Advances in Neural Information Processing Systems* (*NIPS*), 2000, pp. 556–562. ▹ pages 86 and 118.

[91] Q. Li and L. Atlas, "Coherent modulation filtering for speech," in *Proceedings of 2008 IEEE International Conference on Audio, Speech, and Signal Processing* (*ICASSP 2008*), 2008, pp. 4481–4484. ▷ pages 20 and 66.

[92] Y. Li and D. L. Wang, "Separation of singing voice from music accompaniment for monaural recordings," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 15, no. 4, pp. 1475–1487, 2007. ▷ pages 16 and 35.

[93] S.-C. Lim, S.-J. Jang, S.-P. Lee, and M. Y. Kim, "Musig genre/mood classification using afeature-based modulation spectrum," in *Proceedings of International Conference on Mobile IT Convergence*, 2011, pp. 133–136. ▷ page 2.

[94] A. Liutkus, R. Badeau, and G. Richard, "Gaussian process for underdetermined source separation," *IEEE Transactions on Signal Processing*, vol. 50, no. 7, pp. 3155–3167, 2011. ▷ page 97.

[95] A. Loscos, "Spectral processing of the singing voice," Ph.D. dissertation, Universitat Ponpeu Fabra, 2007. ▷ page 6.

[96] H. Lukashevich, M. Gruhne, and C. Dittmar, "Effective singing voice detection in popular music using ARMA filtering," in *Proceedings of Digital Audio Effects 2007* (*DAFx*), 2007. ▷ page 20.

[97] M. Markaki, A. Holzapfel, and Y. Stylianoiu, "Singing voice detection using modulation frequency features," in *Proceedings of Interspeech 2008* (*Interspeech 2008*), 2008, pp. 7–10. ▷ page 20.

[98] A. Mesaros and T. Virtanen, "Automatic recognition of lyrics in singing," *EURASIP Journal on Audio, Speech and Music Processing*, no. 2010, 2010. ▷ pages 2, 6, and 14.

[99] A. Mesaros, T. Virtanen, and A. Klapuri, "Singer identification in polyphonic music using vocal separation and patter recognition methods," in *Proceedings of Internationl Conference on Music Information Retrieval* (*ISMIR 2007*), 2007. ▷ pages 6 and 14.

[100] K. Miyamoto, H. Kameoka, N. Ono, and S. Sagayama, "Separation of Harmonic and Non-Harmonic Sounds Based on Anisotropy in Spectrogram (スペクトログラムの滑らかさの異方性に基づいた調波音・打楽器音の分離)," in *Proceedings of Acoustic Society of Japan Spring Meeting* (日本音響学会講演論文集 (春)), 2008, pp. 903–904, (in Japanese). ▷ pages 25 and 77.

[101] K. Miyamoto, M. Tatezono, J. Le Roux, H. Kameoka, N. Ono, and S. Sagayama, "Separation of harmonic and non-harmonic sounds based on 2D-filtering of the spectrogram (スペクトログラム 2 次元フィルタによる調波音・打楽器音の分離)," in *Proceedings of Acoustic Society of Japan Autumn Meeting* (日本音響学会講演論文集 (秋)), 2007, pp. 825–826, (in Japanese). ▷ pages 25 and 77.

[102] Y. Mizuno, J. Le Roux, N. Ono, and S. Sagayama, "Real-time time-scale/pitch modification of music signal by stretching power spectrogram and consistent phase reconstruction," 2009, pp. 843–844, (in Japanese). ▷ page 121.

[103] Y. Mizuno, "音響信号入出力の自動伴奏のための楽譜追跡と伴奏信号加工アルゴリズムの研究," Master Thesis, The University of Tokyo, 2012, (in Japanese). ▷ pages xvii, 53, 55, 120, and 124.

[104] A. Moreau and A. Flexer, "Drum transcription in polyphonic music using non-negative matrix factorization," in *Proceedings of 6th Internationl Conference on Music Information Retrieval* (*ISMIR 2007*), 2007. ▷ page 118.

[105] H. Mori, W. Odagiri, and H. Kasuya, "F0 dynamics in singing: Evidence from the data of a baritone singer," *IEICE Transactions on Information and Systems*, vol. E87-D, pp. 1086–1092, 2004. ▷ pages 6 and 14.

[106] M. Müller, D. P. W. Ellis, A. Klapuri, and G. Richard, "Signal processing for music analysis," *IEEE Journal of Selected Topics in Signal Processing*, vol. 5, no. 6, pp. 1088–1110, 2011. ▷ pages 4 and 112.

[107] B. Nichols, D. Buttlar, and J. P. Farrell, *Pthreads Programming: A POSIX Standard For Better Multiprocessing.* O'Reilly Nutshell, 1996. ▷ page 62.

[108] J. Ogata and M. Goto, "PodCastle: Collaborative training of acoustic models on the basis of wisdom of crowds for podcast transcription," in *Proceedings of Interspeech 2009* (*Interspeech 2009*), 2009, pp. 1491–1494. ▷ page 3.

[109] Y. Ohishi, M. Goto, K. Itou, and K. Takeda, "Discrimination between singing and speaking voices," *Proceedings of Interspeech 2005* (*Interspeech 2005*), pp. 1141–1144, 2005. ▷ page 74.

[110] ——, "Discrimination between singing and speaking voices using a spectral envelope and a fundamental freuquency derivation," *IPSJ Journal*, vol. 47, no. 6, pp. 1822–1830, 2006, (in Japanese). ▷ page 74.

[111] N. Ono, K. Miyamoto, H. Kameoka, J. Le Roux, Y. Uchiyama, E. Tsunoo, T. Nishimoto, and S. Sagayama, "Harmonic and percussive sound separation and its application to mir-related tasks," in *Advances In Music Information Retrieval*, ser. Studies In Computational Intelligence, Z. W. Ras and A. Wieczorkowska, Eds. Springer, Feb. 2010, pp. 213–236. ▷ page 25.

[112] N. Ono, K. Miyamoto, H. Kameoka, and S. Sagayama, "A real-time equalizer of harmonic and percussive components in music signals," in *Proceedings of 7th Internationl Conference on Music Information Retrieval* (*ISMIR 2008*), 2008, pp. 139–144. ▷ pages xviii, 25, 26, 27, 34, 59, 77, 86, 90, 91, 95, 96, and 98.

[113] N. Ono, K. Miyamoto, J. Le Roux, H. Kameoka, and S. Sagayama, "Separation of a monaural audio signal into harmonic/percussive components by complementary diffusion on spectrogram," in *Proceedings of European Signal Processing Conference 2008* (*EUSIPCO 2008*), 2008. ▷ pages 25 and 77.

[114] A. Ozerov, P. Philippe, F. Bimbot, and R. Gribonval, "Adaptation of bayesian models for single-channel source separation and its application to voice/music separation in popular songs," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 15, no. 5, pp. 1564–1578, 2007. ▷ pages 16 and 35.

[115] A. Ozerov, P. Philippe, R. Gribonval, and F. Bimbot, "One microphone singing voice separation using source-adapted models," in *Proceedings of 2005 IEEE Workshop on Applications of Signal Processing to Audio and Acoustics* (*WASPAA*), 2005, pp. 90–93. ▷ pages 15, 16, and 34.

[116] J. Paulus and T. Virtanen, "Drum transcription with non-negative spectrogram factorization," in *Proceedings of European Signal Processing Conference 2005* (*EUSIPCO 2005*), 2005. ▷ page 118.

[117] G. Poliner, D. P. W. Ellis, A. F. Ehman, E. Goméz, S. Streich, and B. Ong, "Melody transcription from music-audio: Approaches and evaluation," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 15, no. 4, pp. 1247–, 2007. ▷ pages 41 and 42.

[118] L. Rabiner and B.-H. Juang, *Fundamentals of Speech Recognition.* Prentice Hall, 1993. ▷ pages 46 and 122.

[119] S. A. Raczyński, “Harmonic acoustic models and polyphonic probabilistic musicologic models applied to multiple pitch transcription of music signals,” Ph.D. dissertation, The University of Tokyo, 2011. ▷ page 112.

[120] Z. Rafii and B. Pardo, “A simple music/voice separation method based on the extraction of the repeating musical structure,” *Proceedings of 2011 IEEE International Conference on Audio, Speech, and Signal Processing* (*ICASSP 2011*), pp. 221–224, 2011. ▷ pages xiv, 17, 34, 35, and 36.

[121] B. Raj, P. Smaragdis, M. Shashanka, and R. Singh, “Separating a foreground singer from background music,” in *Proc. International Symposium Frontiers of Research Speech and Music* (*FRSM*), 2007. ▷ page 17.

[122] M. Ramona, G. Richard, and B. David, “vocal detection in music with support vector machines,” in *Proceedings of 2008 IEEE International Conference on Audio, Speech, and Signal Processing* (*ICASSP 2008*), 2008. ▷ page 20.

[123] V. Rao and P. Rao, “Melody extraction using harmonic matching,” *Extended Abstract of MIREX*, 2009. ▷ page 50.

[124] ——, “Vocal melody extraction in the presence of pitched accompaniment in polyphonic music,” *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 18, no. 8, pp. 2145–2154, 2010. ▷ pages 2 and 40.

[125] J. Reed, Y. Ueda, S. M. Siniscalchi, Y. Uchiyama, and S. Sagayama, “Minimum classification error training to improve isolated chord recognition,” in *Proceedings of 2009 IEEE International Conference on Audio, Speech, and Signal Processing* (*ICASSP 2009*), 2009, pp. 609–614. ▷ page 25.

[126] L. Regnier and G. Peeters, “Singing voice detection in music tracks using direct voice vibrato detection,” in *Proceedings of 2009 IEEE International Conference on Audio, Speech, and Signal Processing* (*ICASSP 2009*), 2009, pp. 1685–1689. ▷ page 20.

[127] F. Rigaud, M. Lagrange, A. Röbel, and G. Peeters, “Drum extraction from polyphonic music based on a spectro-temporal model of percussive sounds,” in *Proceedings of 2011 IEEE International Conference on Audio, Speech, and Signal Processing* (*ICASSP 2011*), 2011, pp. 381–384. ▷ page 117.

[128] R. T. Rockafellar, *Convex Analysis (Princeton Landmarks in Mathematics And Physics).* Princeton University Press, 1970. ▷ page 113.

[129] S. Russel and P. Novig, *Artificial Intelligence: A Modern Approach, 3rd ed.* Pearson, 2010. ▷ pages 2 and 46.

[130] M. Ryynänen and A. Klapuri, “Transcription of the singing melody in polyphonic music,” *Proceedings of Internationl Conference on Music Information Retrieval* (*ISMIR 2006*), 2006. ▷ pages 2, 17, and 40.

[131] ——, “Automatic transcription of melody, bass line, and chords in polyphonic music,” *Computer Music Journal*, vol. 32, no. 3, 2008. ▷ page 55.

[132] M. Ryynänen, T. Virtanen, J. Paulus, and A. Klapuri, “Accompaniment separation and karaoke application based on automatic melody transcription,” in *Proceedings of 2008 IEEE International Conference on Multimedia and Expo* (*ICME 2008*), 2008, pp. 1417–1420. ▷ page 14.

[133] J. Salamon and Gómez, “Melody extraction from polyphonic music audio,” *Extended Abstract of MIREX*, 2010. ▷ page 50.

[134] J. Salamon and E. Gómez, “Melody extraction from polyphonic music signals using pitch contour characteristics,” *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 20, no. 6, pp. 1759–1770, 2012. ▷ pages 2 and 40.

[135] S. M. Schimmel, “Theory of modulation frequency analysis and modulation filtering, with application to hearing devices,” Ph.D. dissertation, University of Washington, 2007. ▷ page 20.

[136] B. Schuller, A. Lehmann, F. Weninger, F. Eyben, and G. Rigoll, “Blind enhancement of the rhythmic and harmonic sections by NMF: Does it help?” in *In Proc. NAG/DAGA*, 2009, pp. 361–364. ▷ pages xvi, xviii, 95, 96, 99, and 118.

[137] G. A. F. Seber, *A matrix handbook for statisticians.* Wiley Interscience, 2008. ▷ page 114.

[138] P. Smaragdis and J. C. Brown, “Non-negative matrix factorization for polyphonic music transcription,” in *Proceedings of 2003 IEEE Workshop on Applications of Signal Processing to Audio and Acoustics* (*WASPAA*), 2003, pp. 177–180. ▷ pages 17 and 118.

[139] J. Sounders, “Real-time discrimination of broadcast speech / music,” *Proceedings of 1996 IEEE International Conference on Audio, Speech, and Signal Processing* (*ICASSP 1996*), vol. 2, pp. 993–996, 1996. ▷ page 74.

[140] S. S. Stevens, “The measurement of loudness,” *Journal of Acoustic Society of America*, vol. 27, no. 5, pp. 815–829, 1955. ▷ page 80.

[141] S. Sukittanon, L. E. Atlas, and J. W. Pitton, “Modulation-scale analysis for content identification,” *IEEE Transactions on Signal Processing*, vol. 52, no. 10, pp. 3023–3035, 2004. ▷ pages 20 and 66.

[142] J. Sundberg, *The Science of the Singing Voice.* Northern Illinois University Press, 1987. ▷ pages 6 and 14.

[143] M. Suzuki, T. Hosoya, A. Ito, and S. Makino, “Music information retrieval from a singing voice using lyrics and melody information,” *EURASIP Journal on Advances in Signal Processing*, 2007. ▷ pages 2, 6, and 14.

[144] W.-H. Tsai and H.-M. Wang, “Automatic identification of the sung language in popular music recordings,” *Journal of New Music Research*, vol. 36, pp. 105–114, 2007. ▷ pages 6 and 14.

[145] E. Tsunoo, N. Ono, and S. Sagayama, “Rhythm map: Extraction of unit rhythmic patterns and analysis of rhythmic structure from music acoustic signals,” in *Proceedings of 2009 IEEE International Conference on Audio, Speech, and Signal Processing* (*ICASSP 2009*), 2009, pp. 185–188. ▷ page 25.

[146] E. Tsunoo, G. Tzanetakis, N. Ono, and S. Sagayama, “Beyond timbral statistics: Improving music classification using percussive patterns and bass lines,” *IEEE Trans. Audio, Speech and Lang., Process.*, vol. 19, no. 4, pp. 1003–1014, 2011. ▷ page 90.

[147] V. Tyagi, I. McCowan, H. Misra, and H. Bourlard, “Mel-cepstrum modulation spectrum (MCMS) features for robust ASR,” in *IEEE ASRU*, pp. 399–404. ▷ pages 20 and 66.

[148] G. Tzanetakis and P. Cook, “Musical genre classification of audio signals,” *IEEE Transactions on Speech and Audio Processing*, vol. 10, no. 5, pp. 293–302, 2002. ▷ page 2.

[149] Y. Ueda, Y. Uchiyama, T. Nishimoto, N. Ono, and S. Sagayama, “HMM-based approach for automatic chord detection using refined acoustic features,” in *Proceedings of 2010 IEEE International Conference on Audio, Speech, and Signal Processing* (*ICASSP 2010*), 2010, pp. 5518–5521. ▷ pages 2 and 25.

[150] C. Uhle, C. Tiddmar, and T. Sporer, “Extraction of drum tracks from polyphonic music using independent subspace analysis,” in *Proc. Ica*, 2003, pp. 843–847. ▷ pages 117 and 118.

[151] S. Vembu and S. Baumann, “Separation of vocals from polyphonic audio recordings,” in *Proceedings of Internationl Conference on Music Information Retrieval* (*ISMIR 2005*), 2005, pp. 337–344. ▷ page 17.

[152] E. Vincent, “Musical source separation using time-frequency source priors,” *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 14, no. 1, pp. 1273–1282, 2006. ▷ page 118.

[153] E. Vincent, S. Araki, F. J. Theis, G. Nolte, P. Bofill, H. Sawada, A. Ozerov, B. V. Gowreesunker, D. Lutter, and N. Q. K. Duong, “The signal separation evaluation campaign (2007-2010): Achievements and remaining challenges,” *Signal Processing*, vol. 92, pp. 1928–1936, 2012. ▷ page 97.

[154] E. Vincent, R. Gribonval, and C. FéVotte, “BASS-DB: The blind audio source separation evaluation database,” http://www.inria.fr/metiss/bass-db. ▷ pages xiii, 18, and 97.

[155] E. Vincent, R. Gribonval, and C. Févotte, “Performance measurement in blind audio source separation,” *IEEE Transactions on Speech and Audio Processing*, vol. 14, no. 4, pp. 1462–1469, 2006. ▷ pages 93 and 97.

[156] M. Vinyes, “MTG MASS database,” 2008, http://www.mtg.upf.edu/static/mass/resources. ▷ page 97.

[157] T. Virtanen, “Sound source separation using sparse coding with temporal continuity objectives,” in *Proceedings of International Computer Music Conference 2003* (*ICMC 2003*), 2003, pp. 231–234. ▷ page 79.

[158] ——, “Monaural sound source separation by nonnegative matrix factorization with temporal continuity and sparseness criteria,” *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 3, no. 15, pp. 1066–1074, 2007. ▷ page 79.

[159] T. Virtanen, A. T. Cemgil, and S. J. Godsill, “Bayesian extensions to nonnegative matrix factorization for audio signal processing,” in *Proceedings of 2008 IEEE International Conference on Audio, Speech, and Signal Processing* (*ICASSP 2008*), 2008. ▷ pages 79, 118, and 119.

[160] T. Virtanen, A. Mesaros, and M. Ryynänen, “Combining pitch-based inference and non-negative spectrogram factorization in separating vocals from polyphonic music,” in *Proceedings of Workshops on Statistical and Perceptual Audition* (*SAPA*), 2008, pp. 17–20. ▷ pages 17 and 79.

[161] M. Wendelboe, “Using OQSTFT and a modified SHS to detect the melody in polyphonic music (mirex 2009),” *Extended Abstract of MIREX*, 2009. ▷ page 50.

[162] F. Weninger, A. Lehmann, and B. Schuller, “OpenBliSSART: Design and evaluation of a research toolkit for blind source separation in audio recognition tasks,” in *Proceedings of 2011 IEEE International Conference on Audio, Speech, and Signal Processing* (*ICASSP 2011*), 2011, pp. 1625–1628. ▷ pages xvi, xviii, 95, 96, and 99.

[163] *White Paper of Leisure 2011* (レジャー白書 2011 進むレジャーの「デジタル」化). Japan Productivity Center (公益財団法人　日本生産性本部), 2011, (in Japanese). ▷ pages 3, 6, and 53.

[164] *White Paper of Leisure 2012* (レジャー白書 2012 震災後の余暇を考える). Japan Productivity Center (公益財団法人　日本生産性本部), 2012, (in Japanese). ▷ pages 3, 6, and 53.

[165] Z. Xun and F. Tarocco, *Karaoke: The Global Phenomenon.* Reaktion Books, 2007. ▷ pages 6 and 53.

[166] J. Yoo, M. Kim, K. Kang, and S. Choi, "Nonnegative matrix partial co-factorization for drum source separation," in *Proceedings of 2010 IEEE International Conference on Audio, Speech, and Signal Processing* (*ICASSP 2010*), 2010, pp. 1942–1945. ▷ page 118.

[167] K. Yoshii, M. Goto, K. Komatani, T. Ogata, and H. G. Okuno, "Drumix: An audio player with real-time drum-part rearrangement functions for active music listening," *IPSJ Journal*, vol. 48, no. 3, pp. 1229–1239, 2007. ▷ pages 3 and 117.

[168] K. Yoshii, M. Goto, and H. G. Okuno, "Automatic drum sound description for real-world music using template adaptation and matching methods," in *Proceedings of Internationl Conference on Music Information Retrieval* (*ISMIR 2004*), 2004, pp. 184–191. ▷ page 117.

[169] B. Zhu, W. Li, R. Li, and X. Xue, "Multi-stage non-negative matrix factorization for monaural singing voice separation," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 21, no. 10, pp. 2096–2107, Oct 2013. ▷ page 14.

[170] X. Zhu, G. Beauregard, and L. Wise, "Real-time iterative spectrum inversion with look ahead," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 15, no. 5, pp. 1645–1653, 2007. ▷ pages 121 and 124.

[171] A. Zils, F. Pachet, O. Delerue, and F. Gouyon, "Automatic extraction of drum tracks from polyphonic music signals," in *Proceedings of The 2nd International Conference on Web Delivering of Music*, 2002. ▷ page 117.

List of Author’s Publications

1. Papers That are Related to This Dissertation

Journal Papers

[301] Hideyuki Tachibana, Nobutaka Ono, Hirokazu Kameoka, and Shigeki Sagayama, “Harmonic Percussive Sound Separation Based on Anisotropic Smoothness of Spectrograms,” (in review). ▷ Chapter 6

[302] Hideyuki Tachibana, Nobutaka Ono, and Shigeki Sagayama, “Singing Voice Enhancement in Monaural Music Signals Based on Two-stage Harmonic/Percussive Sound Separation on Multiple Resolution Spectrograms,” *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, Vol. 22, No. 1, pp. 228–237 ,2014. ▷ Chapter 2 and 3

Peer-reviewed Conference Papers

[303] Hideyuki Tachibana, Hirokazu Kameoka, Nobutaka Ono, and Shigeki Sagayama, “Comparative Evaluation of Multiple Harmonic/Percussive Sound Separation Techniques based on Anisotropic Smoothness of Spectrogram,” in *Proceedings of 2012 IEEE International Conference on Audio, Speech, and Signal Processing* (*ICASSP 2012*), pp. 465-468, Mar., 2012, Kyoto. ▷ Chapter 6

[304] Hideyuki Tachibana, Nobutaka Ono, and Shigeki Sagayama, “Singing Voice Enhancement for Monaural Music Signals Based on Multiple Time-Frequency Analysis,” in *Proceedings of Intersinging*, Oct., 2010, Tokyo. ▷ Chapter 2 and 3

[305] Hideyuki Tachibana, Takuma Ono, Nobutaka Ono, and Shigeki Sagayama, “Melody Line Estimation in Homophonic Music Audio Signals Based on Temporal-Variability of Melodic Source,” in *Proceedings of 2010 IEEE International Conference on Audio, Speech, and Signal Processing* (*ICASSP 2010*), pp. 425-428, Mar., 2010, Dallas. ▷ Chapter 2 and 3

Technical Reports & Unrefereed Papers

[306] 橘 秀幸, 亀岡 弘和, 小野 順貴, 嵯峨山 茂樹, ”スペクトログラムの長距離滑らかさを考慮した調波打楽器音分離”, in *IPSJ SIG Technical Report* (情報処理学会研究報告) 2013年5月，東京（お茶の水女子大学） ▷ Chapter 6

[307] 橘 秀幸, 水野 優, 小野 順貴, 嵯峨山 茂樹, ”歌声抑圧手法とピッチ変換手法に基づくリアルタイム自動カラオケ生成システム Euterpe,” in Proceedings of Acoustic Society of Japan Autumn Meeting (日本音響学会講演論文集 (秋)), 2-10-11, 2012年9月20日，長野（信州大学） ▷ Chapter 4

[308] 安藤 大地, 阿部 ちひろ, 遠藤 秀行, 伊藤 仁, 伊東 直哉, 三木 翔太, 森勢 将雅, 中村 友彦, 岡田 美咲, 佐々木 将人, 澤田 真吾, 橘 秀幸, 飛世 速光, 宇野 愛, 魚田 知美, 山田 祐雅, 山本 龍一, "デモンストレーション：音楽情報処理の研究紹介 XI," in *IPSJ SIG Technical Report* (情報処理学会研究報告), MUS-96, No. 18, 2012 年 8 月 10 日，金沢

- 橘 秀幸, 水野 優, 小野 順貴, 嵯峨山 茂樹, "Euterpe1.1: 自動カラオケ（マイナスワン）生成システム" ▷ Chapter 4

[309] 橘 秀幸, 亀岡 弘和, 小野 順貴, 嵯峨山 茂樹, "スペクトログラムの滑らかさに基づく調波音・打楽器音分離の各手法の性能比較," in *Proceedings of Acoustic Society of Japan Autumn Meeting* (日本音響学会講演論文集 (秋)), 1-3-14, pp.901-904, 2011 年 9 月 20 日，松江（島根大学） ▷ Chapter 6

[310] 水野 優, 橘 秀幸, 嵯峨山 茂樹, "指揮／演奏追従再生システムのための音楽音響信号の位相再構成法に基づく実時間テンポ変換," in *Proceedings of Acoustic Society of Japan Autumn Meeting* (日本音響学会講演論文集 (秋)), 1-3-12, pp.897-898, 2011 年 9 月 20 日，松江（島根大学） ▷ Chapter 4

[311] 橘 秀幸, 小野 順貴, 嵯峨山 茂樹, "変動長スペクトル特徴量を用いた朗読音声と歌声の判別," in *Proceedings of Acoustic Society of Japan Spring Meeting* (日本音響学会講演論文集 (春)), 1-5-1, 2011 年 3 月 9 日，東京（早稲田大学） ▷ Chapter 5

[312] 橘 秀幸, 小野 順貴, 嵯峨山 茂樹, "周波数-変動長フィルタによる音声強調," 電子情報通信学会技術研究報告, Vol. 110, No.452, SP2010-130, pp. 89–94, (日本音響学会 聴覚研究会資料, Vol. 41, No. 2, H-2011-29, pp.163–168), 2011 年 3 月，東京（東京大学）▷ Chapter 5

[313] 橘 秀幸, 小野 順貴, 嵯峨山 茂樹, "複数スペクトログラムに基づく信号の変動スペクトル表現とそれに基づく信号の新しい分析・分離手法の検討," 第 25 回信号処理シンポジウム, pp. 171-176, 2010 年 11 月, 奈良（奈良女子大学）▷ Chapter 5

[314] 橘 秀幸, 小野 順貴, 嵯峨山 茂樹, "HPSS に基づく音響信号の多重解像度時間周波数解析," in *Proceedings of Acoustic Society of Japan Autumn Meeting* (日本音響学会講演論文集 (秋)), 3-10-5, pp. 607-608, 2010 年 9 月，吹田（関西大学）, 第 2 回学生優秀発表賞 ▷ Chapter 5

[315] 橘 秀幸, 水野 優, 小野 順貴, 嵯峨山 茂樹, "カラオケ (マイナスワン) 演奏生成システム「Euterpe(エウテルペ)」," CrestMuse Symposium 2010, 2010 年 9 月 13 日, 西宮（関西学院大学）▷ Chapter 4

[316] Hideyuki Tachibana, Takuma Ono, Nobutaka Ono, Shigeki Sagayama, Extended abstract for MIREX in Extended Abstract of MIREX, ▷ Chapter 2 and 3

[317] 橘 秀幸, 小野 拓磨, 小野 順貴, 嵯峨山 茂樹, "歌声のゆらぎに着目した歌声強調に基づく音楽音響信号のメロディライン推定," in *Proceedings of Acoustic Society of Japan Autumn Meeting* (日本音響学会講演論文集 (秋)), 2-5-7, pp. 843-844, 2009 年 9 月，郡山（日本大学）▷ Chapter 3

[318] 橘 秀幸, 小野 順貴, 嵯峨山 茂樹, "スペクトルの時間変化に基づく音楽音響信号からの歌声成分の強調と抑圧," in *IPSJ SIG Technical Report* (情報処理学会研究報告), MUS-81, No.12, 2009 年 7 月，福島 ▷ Chapter 2

[319] 橘 秀幸, 小野 順貴, 嵯峨山 茂樹, "多重 HPSS 法による混合音中の音声強調," in *Proceedings of Acoustic Society of Japan Spring Meeting* (日本音響学会講演論文集 (春)), 3-9-7, pp. 721-722, 2009 年 3 月, 東京（東京工業大学）▷ Chapter 5

[320] 橘 秀幸, 小野 順貴, 嵯峨山 茂樹, "多重 HPSS 法によるモノラル音楽音響信号に対するボーカル抑圧," in *Proceedings of Acoustic Society of Japan Spring Meeting* (日本音響学会講演論文集 (春)), 2-8-8, pp. 853-854, 2009 年 3 月, 東京（東京工業大学）▷ Chapter 2

2. Others

Journal Papers

[401] Hideyuki Tachibana, Takafumi Suzuki, and Kunihiko Mabuchi, “Estimating Isometric Tension of Finger Muscle using Needle EMG Signals and the Twitch Contraction Model,” *IEEJ Transactions on Electronics, Information, and Systems*, Vol. 130, No. 2, pp. 254–260, 2010 (in Japanese) （橘 秀幸, 鈴木 隆文, 満渕 邦彦, “針筋電信号と単収縮畳み込みモデルによる手指筋の等尺性収縮時張力の推定,” 電気学会論文誌 C）

Peer-reviewed Conference Papers

[402] Ngoc Q. K. Duong, Hideyuki Tachibana, Emmanuel Vincent, Nobutaka Ono, Rémi Gribonval, and Shigeki Sagayama, “Multichannel Harmonic and Percussive Component Separation by Joint Modeling of Spatial and Spectral Continuity,” in *Proceedings of 2011 IEEE International Conference on Audio, Speech, and Signal Processing* (*ICASSP 2011*), pp. 205-208, May, 2011, Prague.

[403] Takuya Yoshioka, Hideyuki Tachibana, Tomohiro Nakatani, and Masato Miyoshi, “Adaptive Dereverberation of Speech Signals with Speaker-Position Change Detection,” in *Proceedings of 2009 IEEE International Conference on Audio, Speech, and Signal Processing* (*ICASSP 2009*), pp. 3733-3736, Apr., 2009, Taipei.

Technical Reports & Unrefereed Papers

[404] 齋藤 大輔, 石原 達馬, 橘 秀幸, 亀岡 弘和, 嵯峨山 茂樹, ”声質空間上での変換を用いた歌声らしさの転写,” in Proceedings of Acoustic Society of Japan Autumn Meeting (日本音響学会講演論文集 (秋)), 1-2-8, 2012年9月19日，長野（信州大学）

[405] 齋藤 大輔, 石原 達馬, 橘 秀幸, 亀岡 弘和, 嵯峨山 茂樹, ”声質空間上での変換に基づく歌声らしさの転写に関する検討,” in *IPSJ SIG Technical Report* (情報処理学会研究報告), MUS-96, No. 6, 2012年8月9日，金沢

[406] 鈴木 隆文, 橘 秀幸, 戸高 貴文, 古川 浩太郎, 満渕 邦彦, ”針筋電信号による指の動作推定（末梢神経信号による義手制御を目指して）,” 日本機械学会 第22回バイオエンジニアリング講演会, pp. 182, 2010年1月, 岡山（岡山理科大学）

[407] 橘 秀幸, 鈴木 隆文, 満渕 邦彦, ”義手制御に向けた針筋電信号による手指状態推定 - 単収縮モデルを用いた推定手法の提案 - ” 電子情報通信学会技術研究報告, NC2008-56, pp.107-110, (電気学会研究会資料 Vol.MBE-08, pp.53-56), 2008年10月24日, 仙台（東北大学）