Harmonic/Percussive Sound Separation based on Anisotropic Smoothness of Spectrograms

Hideyuki Tachibana, *Member, IEEE,* Nobutaka Ono, *Member, IEEE,* Hirokazu Kameoka, *Member, IEEE,* and Shigeki Sagayama, *Member, IEEE*

***Abstract*—This paper describes a method to separate a monaural music signal into harmonic components e.g. a guitar and percussive components e.g. a snare drum. Separation of these two components is a useful preprocessing for many music information retrieval applications, and in addition, it can be used as a new kind of music equalizer in itself, which enables a music listener to adjust the ratio of the volume of the guitar and the drum freely by themselves. Because of these potential applications, there have been many attempts to develop such a technique, especially in the last decade. However, some of the state-of-the-art techniques have a drawback that they are based on costly operations, such as the multiplications of large-sized matrix, Monte Carlo method, etc., which may constitute barriers to the practical use on some small computers such as smart phones. In this paper, an efficient method that does not depend on these costly operations is described. In formulating the methods, the authors basically assumed only the "anisotropic smoothness" of music spectrogram, which can be one of the minimalistic model that reflects the natures of these instruments. To be specific, the authors just assumed that harmonic instruments are smooth in time, while the percussive instruments are smooth in frequency on a music spectrogram. In this paper, on the basis of the assumption, source separation methods are formulated as optimization problems that optimize the "anisotropic smoothness" under some conditions. Because of the simplicity of the model, the derived algorithms are quite simple. Experimental results show that the methods were effective compared to a state-of-the-art technique, and the computation time was much shorter than an existing method; specifically, it can process a three-minute song in around 4 – 20 seconds on a laptop PC.**

***Index Terms*—audio source separation, music signal processing, harmonic instruments, percussion**

I. Introduction

THIS paper describes a signal separation technique that decomposes an audio music signal into two components: one is a harmonic component such as a guitar, a piano, etc., and the other is a percussive component such as a drum. This paper is an extended version of our previous conference papers [1]–[5].

Music signals, especially commonly distributed popular musics, are often composed of the two different types of musical instruments mentioned. These two typical classes of instruments have very different roles in music. For example, melodies and chords are often played by harmonic components, while a drum beats the tempo. Therefore, when we consider the extraction of music information from music signals, which is one of the major issues of the area of music signal processing, we may naturally assume that rather the former is useful in automatic melody transcription, chord recognition [6], key detection, etc., while the latter is rather useful for automatic tempo estimation [7]. Thus, separating music signals into these two classes of instruments in advance has a significance as a preprocessing for these applications. Indeed, in chord recognition tasks for example, the undesired components, i.e., the percussive components, often act like 'noise,' and interferes with the algorithms' ability to work effectively which result in the poorer performance. In addition, such a technique also has a potential applicability as a music application software, such as a new kind of music player which enables listeners to control the volumes of drum and harmonic sounds separately.

Considering the use of such a technique as a preprocessing and a real-time application, computation efficiency is an important issue. From this motivation, in our previous conference papers [1]–[5], we have proposed an algorithm to separate harmonic and percussive source in music, on the basis of the quite simple concept of "anisotropic smoothness of spectrograms." The key concept of this method is that the directions of the smoothness of the spectrograms of these two typical instruments are different. The spectrograms of harmonic components, such as a guitar, are typically 'smooth' in time, owing to their quasi-stationarity, while the spectrograms of percussive components are typically 'smooth' in the frequency, owing to their impulse-like nature. This concept was first described in [1] in Japanese and in [3] in English, then we reformulated this idea as a MAP estimation problem in [4]. Subsequently, the idea was further extended to stereo signals by multiplying a spatial prior [8], and many other possible formulations are discussed in [5]. In addition to our studies, the concept was followed by some other groups' works, including FitzGerald's median filtering [9]. The applications of HPSS-related techniques for music information retrieval tasks also have been studied [10], including audio chord estimation [6], [11], tempo estimation [7], rhythm map generation [12], [13], and audio melody extraction [14]–[16].

One of the principal advantages of this approach is the simplicity, which results in quite shorter computation time than existing methods (shown in section VII), while the separation performance is almost comparable to an existing method (shown in section VI). This approach is also advantageous in following two points: (1) it is an unsupervised method which requires no pre-training, and (2) it does not require any prior knowledge on the input music signal (for example, it does not need to know exactly what kind of instruments are included in the song).

The main contribution of this paper is that we show the complete descriptions of the concept, formulation (including some refinements to the previous ones in [4], [5]), derivation of algorithms of HPSS, and larger scale evaluations than ever. This paper is organized as follows. In the rest of section I, related work and notations are described.

Manuscript received January 31, 2014.

This work was supported by the JSPS (Japan Society for the Promotion of Science) Grand-In-Aid No. 22-6961. The associate editors coordinating the review of this manuscript and approving it for publication was Dr. xxx.

H. Tachibana was and H. Kameoka is with Graduate School of Information Science and Technology, The University of Tokyo, 7-3-1, Hongo, Bunkyo, Tokyo, 113-8656, Japan.

H. Kameoka is also with NTT Communication Science Laboratory, 3-1, Morinosato Wakamiya, Atsugi, Kanagawa, 243-0198, Japan.

N. Ono is and S. Sagayama was with National Institute of Informatics, 2-1-2, Hitotsubashi, Chiyoda, Tokyo, 101-0003, Japan.

H. Tachibana and S. Sagayama are currently with School of Interdisciplinary Mathematical Sciences, Meiji University, 4-21-1, Nakano, Nakano, Tokyo, 164-8525, Japan. (e-mail: tz14032@meiji.ac.jp, onono@nii.ac.jp, kameoka@hil.t.u-tokyo.ac.jp, sagayama@meiji.ac.jp).

Digital Object Identifier 10.1109/TASLP.2014.2351131

xxxx–xxxx © 2014 IEEE

This is the author's version. Published version is available at https://doi.org/10.1109/TASLP.2014.2351131
© 2014 IEEE. Personal use of this material is permitted. Permission from IEEE must be obtained for all other uses, in any current or future media, including reprinting/republishing this material for advertising or promotional purposes, creating new collective works, for resale or redistribution to servers or lists, or reuse of any copyrighted component of this work in other works.

The principal concept and general formulation of HPSS is described in section II. Explicit formulations based on section II are described in section III. The algorithms to solve the problems are described in section IV. Section V describes examples of the SDR improvement of music signals in the HPSS procedure. Section VI and VII describes the performance evaluations of the proposed methods and relevant harmonic/percussive separation algorithms. Section VIII concludes the paper.

A. Related Work

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 music signal processing area. In order to separate these components, we naturally need to utilize some properties of percussive and harmonic instruments, or, at least some features that are useful to discriminate these instruments (such as a higher order statistics in [17], 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 in order.

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

1) Detection and synthesis e.g. [18]–[25]*, etc.:* This approach first detects specific instruments, typically percussions, and then synthesize a signal based on the detected information (by applying time-frequency masking, for example). In these studies, many features which are useful to discriminate percussive instruments from harmonic instruments are used, such as *phase spectra* [18], *broadbandness of percussive spectra* [22].

2) 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 methods that took this approach was the work by Uhle et al. [17]. They utilized independent subspace analysis (ISA) to decompose a signal. After decomposition, it picks up drum components by a simple decision rule based on the following five features: *Percussiveness*, *Noise-likeness*, *Spectral dissonance* [26], *Spectral flatness measure* (*SFM*) [27], and *Third order cumulant*. The first four criteria model percussive sounds, while the last one does not necessarily model percussive nor harmonic instruments, but is a common criterion in independent component analysis (ICA) [28] and related techniques.

Some other works utilized non-negative matrix factorization (NMF) [29], [30] in a decomposition stage. Helen and Virtanen [31] applied NMF to this problem, in which NMF was followed by SVM. They examined 15 features in all, including 8 spectral features such as *Mel-frequency Cepstral Coefficients* (*MFCC*) and 7 temporal features such as *Periodicity* [32], etc., and verified that some of these features are especially useful in the discrimination. There are still other studies which addressed the relevant approaches, such as the work by Paulus and Virtanen [33], Moreau and Flexer [34], Schuller et al. [35], Yoo et al. [36], and Kim et al. [37], [38].

3) Congregative Signal Decomposition Technique: The third approach considers to develop a novel signal decomposition technique in which some harmonic/percussive discrimination mechanisms are built in. That is, the third approach considers to formulate an algorithm that separates and classifies music signals simultaneously, by using the properties of harmonic and percussive sounds.

Specifically, these methods were formulated like 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. The proposal method is most related to this approach.

Although it did not necessarily address the harmonic and percussive separation problem, an attempt to integrate some properties of music signals into NMF was presented by Virtanen in [39]. In this method, some additional constraint on the gain model, i.e., the temporal structure of the spectrogram, is added. Specifically, *temporal continuity of the gain* and *sparseness of the gain* were incorporated in NMF. Among these, the temporal continuity is quite similar to our temporal smoothness function which shall be described in this paper. (A characteristic of our method is that it has another constraint on percussion which is symmetric to the temporal continuity.)

NMF-based algorithms which principally focused on harmonic and percussion separation have been addressed subsequently. One of the studies of such kind is the work by Vincent et al. [40]. In this study, the model of spectral bases is extended, based on some assumptions such as *spectra of harmonic component is narrowband*, *harmonic instruments have integer overtones*, and *inharmonic components have non-integer overtones*, etc.

It has been pointed out that NMF is well compatible with probabilistic inferences framework [42]. In this perspective, these additional cost functions can be formulated as a prior distributions. On the basis of this concept, many studies have been done in the framework of Bayesian inference or that of MAP estimation. One of the studies which explicitly mentioned the framework is the study of Dikmen and Cemgil [43]. They modeled the prior distribution of the decomposition matrix of NMF taking *temporal (frequency) smoothness of harmonic (percussive) spectrogram* into account. In specific, The prior distribution is modeled by gamma Markov chain, a model of a sequence of positive values [44]. These kinds of distribution is assumed also in another method which is based on tensor factorization, proposed by FitzGerald et al. [45]. These assumptions on harmonic and percussive instruments mentioned in [42], [43], [45] are essentially quite similar to the assumptions of this paper, though the way of formulation is different. (See e.g. Eq. (1), (2), etc.)

A typical drawback, however, of these methods is that they often require much computation resource. For example, in [43], the estimation procedure is based on Gibbs sampling, which is sometimes quite costly. One of our motivations in this paper came from a question how can these congregative approaches be simplified. In our observations, what make these state-of-the-art techniques computationally costly are (1) the multiplications of big matrices which appear in NMF algorithms, and (2) the sampling from a probability density function with rather complicated forms. On the basis of these motivations, we formulated a problem that is *not* based on the NMF framework nor any complicated probability density functions. Instead of them, we simply formulated the problem as an optimization problems based only on a simple assumption *"anisotropic smoothness" of harmonic and percussive spectrogram* and some subsidiary constraints. As a result, the derived algorithms were quite simpler, and they required quite small computation cost. This paper gives the whole formulations of the simple separation algorithms.

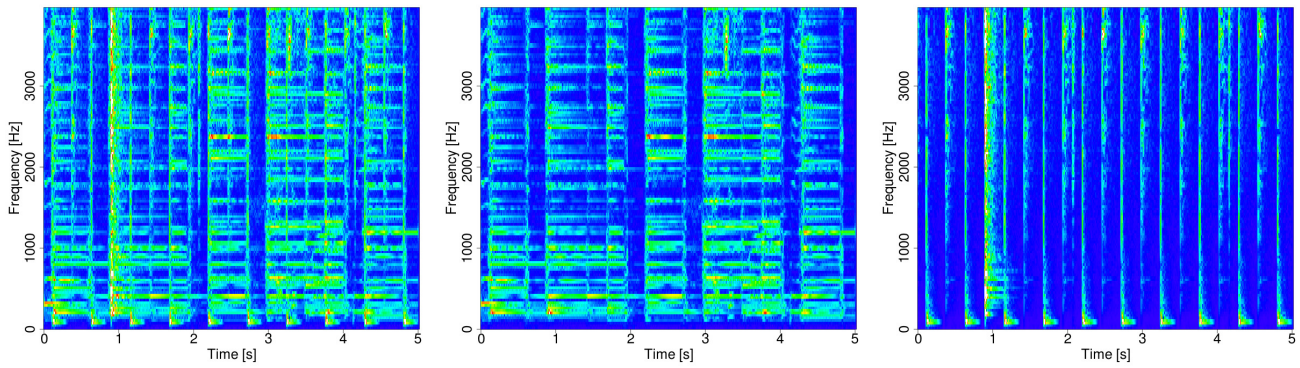

Fig. 1. Spectrograms of (a) a mixed music signal $\boldsymbol{Y}$ (extracted from a song "Another Dreamer – Dreams" in BASS-dB dataset [41]), (b) a harmonic component $\boldsymbol{H}$, in which it is observed that it is rather continuous in time than in frequency, and (c) a percussive component $\boldsymbol{P}$, which is continuous in frequency.

B. Notation

The notation in this paper is as follows. Given a discretized real-valued signal $x(t)$, where t denotes discrete time $t \in \mathbb{Z}$, and let $\hat{\boldsymbol{X}} = \mathsf{STFT}(x(t)) = (\hat{X}_{n,k}) \in \mathbb{C}^{N\times K}$ be a complex spectrogram of the signal, which is derived by applying the short-time Fourier transform (STFT) to $x(t)$. The subscripts $n, k \in \mathbb{Z}$ 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." Clearly the spectrogram above is composed of $N \times K$ bins, where N is the number of time frames, and K is the number of bins in a single frame. Note there is a relation between K and the size of STFT analyzing frame of length L that $K = L/2 + 1$ when L is even. 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 $(n, k) \notin [0, N-1] \times [0, K-1]$.

Unlike NMF-based methods, we do not particularly regard a spectrogram $\hat{\boldsymbol{X}}$ as a matrix in this paper. Instead, we regard it just a tuple of $N \times K$ complex/real numbers, and define arithmetic operations as element-wise operations. For example, $\boldsymbol{XY} := (X_{n,k}Y_{n,k})$, $|\boldsymbol{X}^\gamma|/2 := (|X^\gamma_{n,k}|/2)$, $\boldsymbol{X} \geq 0 \overset{\text{def}}{\Leftrightarrow} \forall (n,k), X_{n,k} \geq 0$, etc. Using these notations, the amplitude spectrogram is denoted as $|\hat{\boldsymbol{X}}| \in \mathbb{R}^{N\times K}$ and the phase spectrogram is written as $\hat{\boldsymbol{X}}/|\hat{\boldsymbol{X}}|$.

In this paper, we often use γ-powered variables "x^γ" instead of "x" For example, $(\partial/\partial(x^\gamma))x^{m\gamma} = (\partial/\partial y)y^m = my^{m-1} = mx^{(m-1)\gamma}$. We often use the expressions such as $x^\gamma \leftarrow f(x^\gamma)$, which are in principle equivalent to $x \leftarrow \sqrt[\gamma]{f(x^\gamma)}$. These expressions mean that we should evaluate $y \leftarrow f(y)$ using a temporary variable y, and evaluate $x \leftarrow \sqrt[\gamma]{y}$ only when the very x becomes needed.

We often omit some arguments of functions if evident. For example, $S(\boldsymbol{H}, \boldsymbol{P}; w)$ is sometimes written as S for simplicity.

II. Concept of HPSS: Anisotropic Smoothness of Spectrogram

A. Concept of 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 Fig. 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 Fig. 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 Fig. 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,

$$H_{n,k} \approx H_{n\pm 1,k}. \tag{1}$$

Note, not entirely identical but essentially similar properties are supposed on harmonic instruments in [18], [39], [42], [43], [45]–[47], etc.

This assumption is generalized as follows,

$$H_{n,k} \approx H_{n\pm n',k}, \quad (1 \leq n' \leq N'), \tag{2}$$

where N' is the maximal distance we consider neighbour. The value of N' is supposed to be from 1 to several dozen, from the observations on the spectrogram Fig. 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, 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 Fig. 1 (c), because of their impulse-like nature. To be specific, the spectrogram of a percussion $\boldsymbol{P} \in \mathbb{R}^{N\times K}$ should have following property similarly to (2),

$$P_{n,k} \approx P_{n,k\pm k'}, \quad (1 \leq k' \leq K') \tag{3}$$

where K' is the maximal distance under consideration.

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 the proposed methods.

B. 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 (2) and (3) are satisfied.

We first define the quantitative criteria on the anisotropic smoothness of a 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,

$$S_{\mathsf{time}}(n,k,\boldsymbol{H}^{\gamma}) := \frac{1}{N'}\sum_{n'=1}^{N'}(H^{\gamma}_{n,k}-H^{\gamma}_{n-n',k})^2, \tag{4}$$

$$S_{\mathsf{freq}}(n,k,\boldsymbol{P}^{\gamma}) := \frac{1}{K'}\sum_{k'=1}^{K'}(P^{\gamma}_{n,k}-P^{\gamma}_{n,k-k'})^2, \tag{5}$$

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 [48], it would be considerable to set the value around 0.6.

When the condition (2) 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 (5). In summary,

$$\text{(2) is satisfied} \approx S_{\mathsf{time}}(n,k,\boldsymbol{H}^{\gamma}) \text{ is small,} \tag{6}$$

$$\text{(3) is satisfied} \approx S_{\mathsf{freq}}(n,k,\boldsymbol{P}^{\gamma}) \text{ is small.} \tag{7}$$

Using these functions that indicate anisotropic smoothness around a single bin, let us define a "total smoothness" functions, that indicate the smoothness of the whole spectrogram. Specifically, we defined them by simple summations of the values of $N \times K$ anisotropic smoothness criteria as follows,

$$S^{\mathsf{total}}_{\mathsf{time}}(\boldsymbol{H}^{\gamma}) := \sum_n\sum_k S_{\mathsf{time}}(n,k,\boldsymbol{H}^{\gamma}) \tag{8}$$

$$S^{\mathsf{total}}_{\mathsf{freq}}(\boldsymbol{P}^{\gamma}) := \sum_n\sum_k S_{\mathsf{freq}}(n,k,\boldsymbol{P}^{\gamma}). \tag{9}$$

C. Validity of the Criteria

Let us verify the validity of the criteria (8) and (9), using real instrumental sounds. The audio files we used for the evaluation were excerpted from RWC-MDB-I-2001 [49] database. Fig. 2 shows the values of $S^{\mathsf{total}}_{\mathsf{time}}(\boldsymbol{X}^{\gamma})$ defined by (8) and $S^{\mathsf{total}}_{\mathsf{freq}}(\boldsymbol{X}^{\gamma})$ defined by (9) for each instrument, where $\boldsymbol{X}^{\gamma}$ is a spectrogram such as a piano, a harpsichord, etc. It is shown that harmonic instruments make (8) small, (9) large, and (8) $\ll$ (9) is satisfied. To the contrary, percussive instruments make (8) relatively large, and (9) relatively small. The fact indicates that it is reasonable to use (8) and (9) as indicators to measure the anisotropic smoothness of the spectrogram $\boldsymbol{X}^{\gamma}$ in time and in frequency, respectively.

D. Smoothness Function and Abstract Formulation of Optimization Problems

We finally define an integration of two criteria (8) and (9) as follows,

$$S(\boldsymbol{H}^{\gamma},\boldsymbol{P}^{\gamma};w) := S^{\mathsf{total}}_{\mathsf{time}}(\boldsymbol{H}^{\gamma}) + wS^{\mathsf{total}}_{\mathsf{freq}}(\boldsymbol{P}^{\gamma}), \tag{10}$$

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 the objective functions of our early studies [3]–[5] when $N' = K' = 1$.

Given a spectrogram $\boldsymbol{Y}^{\gamma}$, and let us consider a thought experiment to make the thus defined S as small as possible by classifying each element $Y^{\gamma}_{n,k}$ into either $\boldsymbol{H}^{\gamma}$ or $\boldsymbol{P}^{\gamma}$, under the condition that we are given information whether a bin $Y^{\gamma}_{n,k}$ 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.*

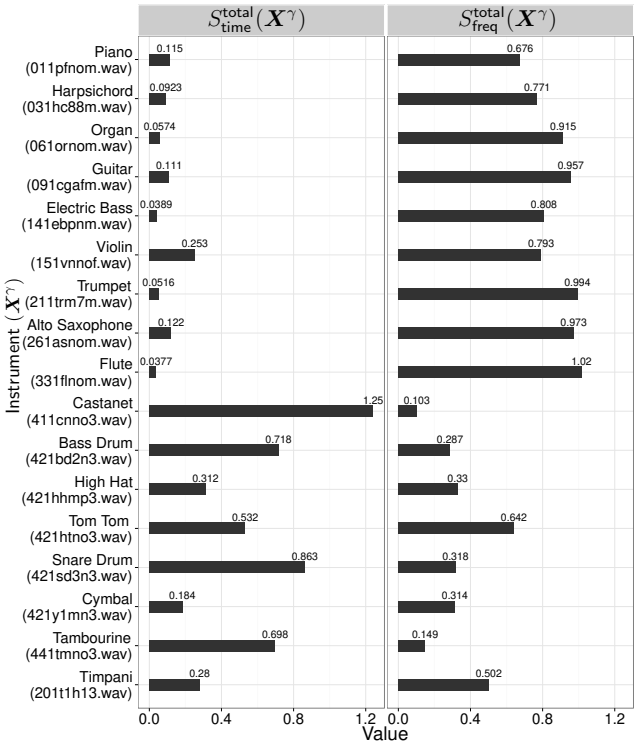

Fig. 2. Values of smoothness functions $S^{\mathsf{total}}_{\mathsf{time}}(\boldsymbol{X}^{\gamma})$ and $S^{\mathsf{total}}_{\mathsf{freq}}(\boldsymbol{X}^{\gamma})$. All the values are normalized by the length and the averaged power of each clip. The condition is as follows: $\gamma = 0.5, N' = K' = 5, L = 1024/16000$ [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 [49]. The figure indicates that temporal smoothness function $S^{\mathsf{total}}_{\mathsf{time}}(\boldsymbol{X}^{\gamma})$ defined by (8) 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 $S^{\mathsf{total}}_{\mathsf{freq}}(\boldsymbol{X}^{\gamma})$ defined by (9) are quite large for harmonic sounds, while they are rather small for percussive sounds.

To return from the digression, what we consider in this paper 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 paper considers the separation of two components with the 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 such 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):
$$\begin{aligned}&\text{minimize} && S(\boldsymbol{H}^{\gamma},\boldsymbol{P}^{\gamma};w) + \text{additional cost}\\ &\text{subject to} && \text{some constraints}\end{aligned}$$

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

$$\boldsymbol{H}^{\gamma} \geq 0, \boldsymbol{P}^{\gamma} \geq 0, \tag{11}$$

because "amplitude" $H_{n,k}$ cannot be negative, which implies the non-

negativity of $H^{\gamma}_{n,k}$.

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.

III. Formulation of HPSS based on the Anisotropic Smoothness

This section describes the explicit formulations of the 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.

A. 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,

$$\boldsymbol{H}^{\xi} + \boldsymbol{P}^{\xi} = \boldsymbol{Y}^{\xi}, \tag{12}$$

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

1) With regard to ξ, any one of the following three assumptions should be assumed, in order to make the condition (12) physically meaningful.
   a) In order to assume $\xi = 1$, we should accept the assumption of the additivity of the amplitude spectrograms. Note, it 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 condition that $\hat{\boldsymbol{H}}$, $\hat{\boldsymbol{P}}$, and $\hat{\boldsymbol{Y}}$ have the identical phase spectrogram, i.e., $H_{n,k}e^{\sqrt{-1}\theta} + P_{n,k}e^{\sqrt{-1}\theta} = Y_{n,k}e^{\sqrt{-1}\theta}$.
   b) In order to assume $\xi = 2$, we should accept the assumption of the additivity of the power spectrograms.
   c) The values other than $\xi = 1, 2$ may also be acceptable, if we assume that $\boldsymbol{H}$ and $\boldsymbol{P}$ rarely share the same bins, but they are mostly exclusive, i.e.,
   $$(H_{n,k}, P_{n,k}) = (Y_{n,k}, 0) \text{ or } (0, Y_{n,k}) \tag{13}$$
   is satisfied in many bins. Note, this assumption is equivalent to "there exists an ideal binary mask that separates the harmonic and the percussive components almost perfectly."
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:
$$\begin{aligned} \text{minimize} \quad & S(\boldsymbol{H}^{\gamma}, \boldsymbol{P}^{\gamma}; w) \\ \text{subject to} \quad & \boldsymbol{H}^{2\gamma} + \boldsymbol{P}^{2\gamma} = \boldsymbol{Y}^{2\gamma}, \\ & \boldsymbol{H}^{\gamma} \geq 0, \quad \boldsymbol{P}^{\gamma} \geq 0. \end{aligned}$$

Note this setting is identical to "HM2" in [5] when $N' = K' = 1$.

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 problem shall be described in Section IV.

B. 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 to add another cost term on the difference between them to the objective function, and derive an algorithm that minimizes the 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 the generalized Kullback-Leibler (KL) divergence, which is one of the basic statistical criterion that has been used in many fields including NMF [29], [50] in order to measure the "distance" between two distributions. Hereafter we just call it the KL divergence. The KL divergence of $\boldsymbol{Y}^{\xi}$ from $\boldsymbol{H}^{\xi} + \boldsymbol{P}^{\xi}$ is defined by

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

where ξ should hold $\xi = 2\gamma$ because of the requirement for the homogeneity of the objective function (see Appendix of [16]). Using the 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 our previous study [4] when $N' = K' = 1$. An algorithm to solve this optimization problem shall be shown in the next section.

IV. Derivation of the Algorithms

In this section, we consider to derive 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.

A. Optimization algorithm for Problem 1-A (HPSS 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.

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

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

$$\mathcal{L} := S(H^{\gamma}_{n,k}, P^{\gamma}_{n,k} | \dots) + \lambda(H^{2\gamma}_{n,k} + P^{2\gamma}_{n,k} - Y^{2\gamma}_{n,k}), \tag{14}$$

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

$$H^{\gamma}_{n,k} = (2+\lambda)^{-1} H^{(n\text{-mean})}_{n,k}, \quad P^{\gamma}_{n,k} = (2w+\lambda)^{-1} P^{(k\text{-mean})}_{n,k} \tag{15}$$

where

$$H^{(n\text{-mean})}_{n,k} := \frac{1}{2N'} \sum_{n'=1}^{N'} (H^{\gamma}_{n+n',k} + H^{\gamma}_{n-n',k}), \tag{16}$$

$$P^{(k\text{-mean})}_{n,k} := \frac{1}{2K'} \sum_{k'=1}^{K'} (P^{\gamma}_{n,k+k'} + P^{\gamma}_{n,k-k'}), \tag{17}$$

which indicate the moving averages of the time-frequency bins around (n,k), excluding (n,k) itself. By substituting $H^{\gamma}_{n,k}$ and $P^{\gamma}_{n,k}$ in $\partial\mathcal{L}/\partial\lambda = 0$ by (15), a quartic equation on λ is derived as follows,

$$\left(\frac{H^{(n\text{-mean})}_{n,k}}{2+\lambda}\right)^2 + \left(\frac{P^{(k\text{-mean})}_{n,k}}{2w+\lambda}\right)^2 = Y^{2\gamma}_{n,k}. \tag{18}$$

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,

$$(\lambda+2)^2 = \frac{1}{Y^{2\gamma}_{n,k}} \left\{ (H^{(n\text{-mean})}_{n,k})^2 + (P^{(k\text{-mean})}_{n,k})^2 \right\}. \tag{19}$$

Using thus obtained λ, and noting that $H^{\gamma}_{n,k}$ and $P^{\gamma}_{n,k}$ should be positive, the equations on extrema are derived as follows[1],

$$H^{\gamma}_{n,k} = \frac{H^{(n\text{-mean})}_{n,k}}{\sqrt{(H^{(n\text{-mean})}_{n,k})^2 + (P^{(k\text{-mean})}_{n,k})^2}} Y^{\gamma}_{n,k}, \tag{20}$$

$$P^{\gamma}_{n,k} = \frac{P^{(k\text{-mean})}_{n,k}}{\sqrt{(H^{(n\text{-mean})}_{n,k})^2 + (P^{(k\text{-mean})}_{n,k})^2}} Y^{\gamma}_{n,k}. \tag{21}$$

We can use (20) and (21) as a tentative solution for a single time-frequency bin (n,k). That is,

$$H^{\gamma}_{n,k} \leftarrow \text{r.h.s. of (20)}, \tag{22}$$

$$P^{\gamma}_{n,k} \leftarrow \text{r.h.s. of (21)}. \tag{23}$$

This substitution *decreases* (precisely, *does not increase*) the objective function S. Moreover, it is evident that applying the same as above to all time-frequency bins will never increase S.

By summarizing the discussion above and filling in with details, the whole procedure is written as follows.

[1]Note that we can obtain an algorithm which is identical to FitzGerald's median filtering [9], just by replacing $\cdot$-mean by $\cdot$-median, which is defined by

$$H^{(n\text{-median})}_{n,k} := \text{median}((H_{n+n',k})_{-N' \le n' \le N'}),$$

etc. This fact intuitively implies that we can roughly interpret the FitzGerald's method as a technique that approximately minimizes an absolute-value-based cost function $S' = \sum_n \sum_k \sum_{n'} |H_{n,k} - H_{n+n',k}| + \cdots$, noting that the median $m = \text{median}(\{x_i\})$ minimizes $\sum_i |x_i - m|$ where $\{x_i\}_i$ is a set of samples. In light of the fact mentioned, we can consider a general "center value" m, which is characterized as the minimizer of the "variance" induced by a distance measure $d(\cdot,\cdot)$, i.e., $m = \text{argmin}_{\mu} \sum_{i=1}^{n} d(x_i,\mu)^2$. These centers are sometimes called "Fréchet mean" [51]. The means in this class may also be exploited in the literature of HPSS in the future.

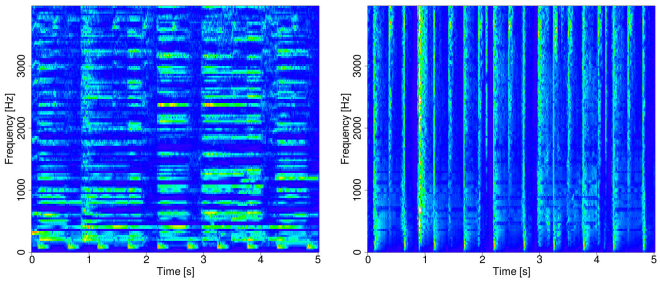

Fig. 3. An example of the result of Method 1-A. Number of Iteration was $I = 5$, and the parameters are $N' = K' = 8$. The left is harmonic and the right is percussive. The input signal is Fig. 1(a), and the ground truths are Fig. 1(b), (c)

Method 1-A (HPSS 1-A):

i. Given a complex spectrogram $\hat{\boldsymbol{Y}} \in \mathbb{C}^{N\times K}$, and take its absolute value $\boldsymbol{Y} = |\hat{\boldsymbol{Y}}| \in \mathbb{R}^{N\times K}$.
ii. 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 paper.
iii. Update $\boldsymbol{H}^{\gamma}$ and $\boldsymbol{P}^{\gamma}$ using (22) and (23).
iv. Iterate iii for I times.
v. 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., $x(t) = \mathsf{STFT}^{-1}[\boldsymbol{X}\hat{\boldsymbol{Y}}/\boldsymbol{Y}]$.

Note, despite the convexity[2] of the objective function S, the problem is not convex programming, because the equality constraints $H^{2\gamma}_{n,k} + P^{2\gamma}_{n,k} = Y^{2\gamma}_{n,k}$ does not satisfy the requirement that the equation constraints should be affine (first-degree equations). Therefore, just decreasing the objective function does not necessarily result in a global minimum, but the solution typically falls in a local minimum, depending on the initial value. There are some meta-heuristics to find a better solution in these kinds of optimization problems, such as testing many initial values, applying genetic algorithms, etc. Nevertheless, we avoided these rather costly operations that contradict our original purpose of developing an efficient source separation technique. Instead, we simply set the initial values as the input spectrogram $\boldsymbol{Y}$, as shown in (ii). It may not be necessarily the best initial value, but it empirically performed well in our preliminary experiments.

Fig. 3 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, compared to the mixed signal, Fig. 1 (a).

B. Optimization algorithm for Problem 1-B (HPSS 1-B)

We may similarly derive the optimization procedure for Problem 1-B. The updating formulae are written as follows[3].

$$H^{\gamma}_{n,k} \leftarrow \rho(\alpha_{n,k}; 0, Y^{\gamma}_{n,k}) \tag{24}$$

$$P^{\gamma}_{n,k} \leftarrow \rho(\beta_{n,k}; 0, Y^{\gamma}_{n,k}) \tag{25}$$

where

$$\alpha_{n,k} = \frac{1}{2}\left(Y^{\gamma}_{n,k} + H^{(n\text{-mean})}_{n,k} - P^{(k\text{-mean})}_{n,k}\right) \tag{26}$$

$$\beta_{n,k} = \frac{1}{2}\left(Y^{\gamma}_{n,k} - H^{(n\text{-mean})}_{n,k} + P^{(k\text{-mean})}_{n,k}\right) \tag{27}$$

[2]It is easily verified that Hessian of S w.r.t. $H^{\gamma}_{n,k}$ and $P^{\gamma}_{n,k}$ is positive semi-definite.

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

$$\rho(x;l,u)=\begin{cases} l & \text{if } x<l \\ x & \text{if } l\le x\le u \\ u & \text{if } u<x. \end{cases} \tag{28}$$

This substitution *decreases* (precisely, *does not increase*) the objective function S by the same reason above. Moreover, Problem 1-B is a convex programming, which implies that a local optimum is always a global optimum.

C. Optimization algorithm for Problem 2 (HPSS 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 minimizes $U(H_{n,k}^{\gamma}|\boldsymbol{H}^{\gamma},\boldsymbol{P},\boldsymbol{Y})$ w.r.t. the variable $H_{n,k}^{\gamma}$ should hold $\partial U/\partial(H_{n,k}^{\gamma})=0$. Therefore, applying the updating formulae that are derived by solving this equation does not increase U. It is not difficult to solve this equation theoretically, but inconveniently, it result in cubic equations on $H_{n,k}^{\gamma}$, which are costly to solve. We then consider to simplify the problem, using the following trick, which reduces the degree of the problem to quadratic. The idea of the following discussion is based on the techniques that were used in some other studies such as [52]. Note that the Appendix of [13] also describes a similar discussion.

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 the 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,

$$\begin{aligned} &D_{\mathsf{KL}}(\boldsymbol{Y}^{2\gamma}\|\boldsymbol{H}^{2\gamma}+\boldsymbol{P}^{2\gamma})\le \\ &\quad D_{\mathsf{KL}}(\boldsymbol{\theta}\boldsymbol{Y}^{2\gamma}\|\boldsymbol{H}^{2\gamma})+D_{\mathsf{KL}}((\boldsymbol{1}-\boldsymbol{\theta})\boldsymbol{Y}^{2\gamma}\|\boldsymbol{P}^{2\gamma}). \end{aligned} \tag{29}$$

This inequality is easily proved using the following inequality,

$$-\ln(x+y)\le -\theta\ln\frac{x}{\theta}-(1-\theta)\ln\frac{y}{1-\theta}, \tag{30}$$

where $x,y>0, 0<\theta<1$. The equality of (29) is satisfied only when

$$\theta_{n,k}=\frac{H_{n,k}^{2\gamma}}{H_{n,k}^{2\gamma}+P_{n,k}^{2\gamma}}=:\theta_{n,k}^{\star}. \tag{31}$$

The inequality (29) yields the following inequality,

$$\begin{aligned} U &= S(\boldsymbol{H}^{\gamma},\boldsymbol{P}^{\gamma};w)+\mu D_{\mathsf{KL}}(\boldsymbol{Y}^{2\gamma}\|\boldsymbol{H}^{2\gamma}+\boldsymbol{P}^{2\gamma}) \\ &\le S(\boldsymbol{H}^{\gamma},\boldsymbol{P}^{\gamma};w) \\ &\quad +\mu\{D_{\mathsf{KL}}(\boldsymbol{\theta}\boldsymbol{Y}^{2\gamma}\|\boldsymbol{H}^{2\gamma})+D_{\mathsf{KL}}((\boldsymbol{1}-\boldsymbol{\theta})\boldsymbol{Y}^{2\gamma}\|\boldsymbol{P}^{2\gamma})\} \\ &=: U^{+}(\boldsymbol{H}^{\gamma},\boldsymbol{P}^{\gamma},\boldsymbol{\theta};\boldsymbol{Y}^{\gamma},w,\mu), \end{aligned} \tag{32}$$

where U^{+} is an auxiliary function that gives an upper bound of U.

From here, we tentatively consider to derive a sequence that decreases U^{+} instead of U. The original purpose to obtain a sequence that decreases 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 results in the following quadratic equations,

$$a_1(H_{n,k}^{\gamma})^2-2b_1H_{n,k}^{\gamma}-c_1=0, \tag{33}$$

$$a_2(P_{n,k}^{\gamma})^2-2b_2P_{n,k}^{\gamma}-c_2=0, \tag{34}$$

where

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

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^{+}, the following updating formulae are obtained[4],

$$H_{n,k}^{\gamma}\leftarrow\frac{b_1+\sqrt{b_1^2+a_1c_1}}{a_1}, \tag{35}$$

$$P_{n,k}^{\gamma}\leftarrow\frac{b_2+\sqrt{b_2^2+a_2c_2}}{a_2}. \tag{36}$$

In addition to (35) and (36), we can consider the minimization of U^{+} w.r.t. $\theta_{n,k}$. Clearly the $\theta_{n,k}$ that makes U^{+} minimal is none other than $\theta_{n,k}^{\star}$ in (31), because

$$U^{+}(\theta_{n,k}=\theta_{n,k}^{\star})=U\le U^{+}(\theta_{n,k}). \tag{37}$$

(Note all variables except $\theta_{n,k}$ are fixed here, and U is independent of $\theta_{n,k}$.) Therefore, substituting $\theta_{n,k}$ by the r.h.s of (31) also *decreases* (*does not increase*) U^{+}.

In the updating procedure (35), (36) and (31), 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 follows.

Method 2 (HPSS 2):

i. Given a complex spectrogram $\hat{\boldsymbol{Y}}$, and take $\boldsymbol{Y}=|\hat{\boldsymbol{Y}}|$
ii. Set initial values to $\boldsymbol{H}^{\gamma}=\boldsymbol{P}^{\gamma}=\boldsymbol{Y}^{\gamma}/\sqrt{2}$, similarly to HPSS 1-A.
iii. Update $\boldsymbol{H}^{\gamma}$, $\boldsymbol{P}^{\gamma}$ and $\boldsymbol{\theta}$ using (35), (36) and (31).
iv. Iterate iii for I times.
v. Apply some postprocessings to $\boldsymbol{H}$ and $\boldsymbol{P}$. We applied Wiener mask in this paper.
vi. Apply inverse STFT in order to obtain $h(t)$ and $p(t)$ similarly to Problem 1.

Since we did not lay strict constraints 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 (v 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 the limit case of the penalty function method.

V. Dependencies of Performance on $M:=N'=K'$

In this section, we specify the value of N' and K' in S, which are defined in around equations (2), (3) as the maximal distances on spectrogram that we consider neighbour in temporal direction (N') and frequency direction (K'), respectively. We consider the case $N'=K'=:M$ for simplicity, though N' and M' can be tuned independently if needed.

A. Evaluation of Computation Time of Single Update

We first evaluated the computation time of applying the 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™i7 CPU Q 740 @ 1.73GHz, and the OS was Linux on the VMware. In evaluating the computation

[4]Note these equations are identical to our previous studies [4], when $w=1,\mu=2\sigma_{\mathrm{H}}^2=2\sigma_{\mathrm{P}}^2,N'=K'=1$.

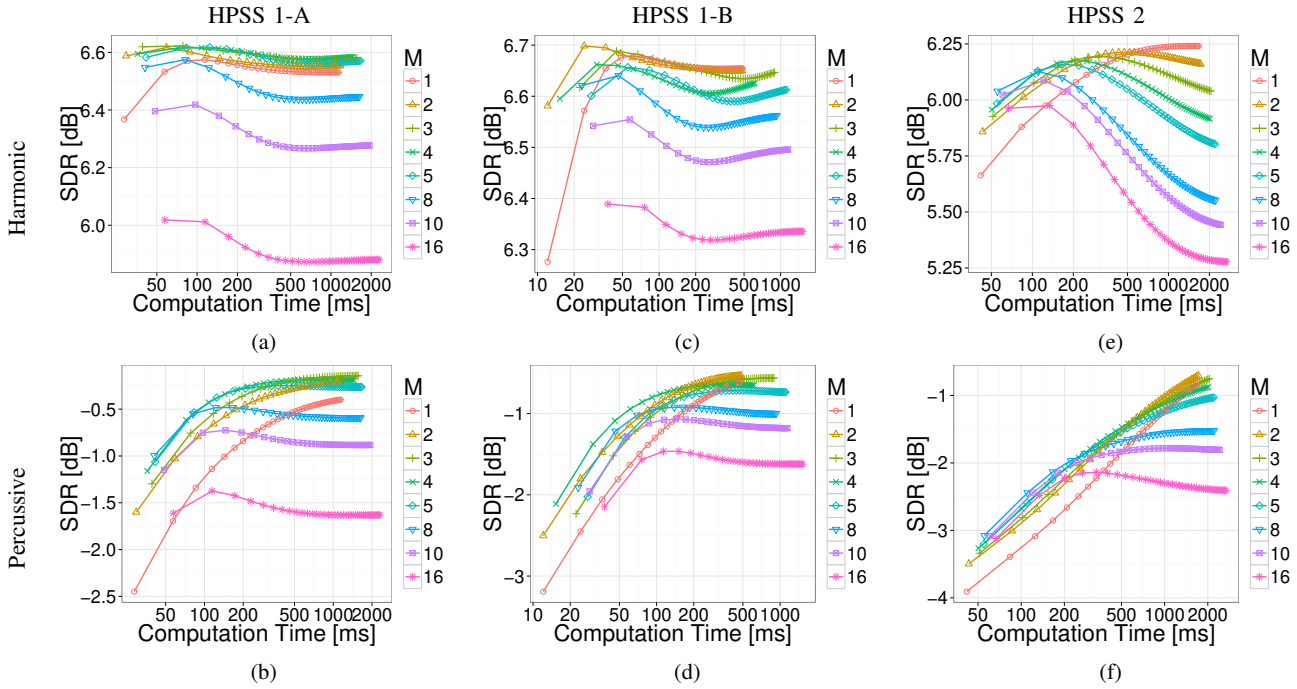

Fig. 4. Relation to computation time (log scale) and SDR (averaged value of 6 songs in MASS database [53]) of the output Harmonic and Percussive signals, obtained by each HPSS method. (a), (b) HPSS 1-A, (c), (d) HPSS 1-B, and (e), (f) HPSS 2. M denotes $M := N' = K'$. Each point indicates the ending time of each iteration. Note computation time was not directly evaluated, but calculated from the results of Section V-A (Table I) using (38), i.e., the computation time indicated in the figure is for the 16-kHz 20-second music signals.

TABLE I
COMPUTATION TIME OF APPLYING UPDATING FORMULAE TO A 16-KHZ 20-SECOND INPUT SIGNAL ONCE, $T_{\text{single\_update\_20[s]}}(M)$, *ESTIMATED* BY AVERAGING 100 TIMES ITERATIONS. UNIT: [ms].

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

time, we applied the updating formulae 100 times, instead of applying them once, and we divided the measured computation time by 100. Hereafter we shall denote the obtained computation time as $T_{\text{single\_update\_20[s]}}(M)$.

Table I shows the computation time of each HPSS algorithm's single update $T_{\text{single\_update\_20[s]}}(M)$, 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 the square root, which are much more costly than computation of $H_{n,k}^{(n\text{-mean})}$, are 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.

B. Examples of SDR improvement in HPSS updating

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

The data we used here were the 6 songs in the MASS database [53]. The length of each song was around 10 [s], and the sampling rate was 16 kHz. We mixed them in 5 dB (Harmonic to Percussive ratio).

Figure 4 shows the result: the SDR [54] curves of the harmonic and percussive components for each M. The SDR indicate the averaged values of those of the 6 songs. The computation time was not measured directly, but was estimated by the following formula.

$$\text{Computation Time}(I, M) = T_{\text{single\_update\_20[s]}} \times I \qquad (38)$$

That is, it should be noted that the computation time displayed in the figures indicate the time which will be required to process a 20-second 16-kHz music signal.

We may observe that M around 2 to 5 improve SDR faster than the others. Updating too much do not necessarily result in the better separation performance, which may be because $\boldsymbol{H}$ and $\boldsymbol{P}$ "overfit" to our too simple model. 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 II, III, and IV; the values (a) emphasized rather on the separation performance, and the values (b) on the efficiency. Of course, we may also consider other values for specific applications, but we only consider the two representatives for simplicity in this paper.

VI. COMPARATIVE EVALUATIONS OF PROPOSED AND EXISTING METHODS USING PROFESSIONALLY-CREATED MUSIC SIGNALS

A. Parameter Setting of HPSS 1-A, 1-B, and 2

The parameters are shown in Table II, III, IV. We considered two parameter sets (a) and (b) for each method. The value of w was decided to evenly weight the cost on harmonic and percussive components. The value of μ was empirically decided, taking our previous studies into account. N' and M' are decided on the basis of the results of the previous section.

TABLE II
PARAMETER SETTINGS OF HPSS 1-A

Emphasis on	Value (a) SDR	Value (b) time
Range $M = N' = K'$	4	2
Number of iteration I	10	2
Exponential factor γ	0.5	0.5
Frame length of STFT L	1024 (64 [ms])	1024 (64 [ms])
Frame shift of STFT s	256 (16 [ms])	256 (16 [ms])
Window Function	hanning	hanning

TABLE III
PARAMETER SETTINGS OF HPSS 1-B

Emphasis on	Value (a) SDR	Value (b) time
Range $M = N' = K'$	4	2
Number of iteration I	10	2
Exponential factor γ	0.5	0.5
Frame length of STFT L	1024 (64 [ms])	1024 (64 [ms])
Frame shift of STFT s	256 (16 [ms])	256 (16 [ms])
Window Function	hanning	hanning

TABLE IV
PARAMETER SETTINGS OF HPSS 2 (THE PARAMETERS w, μ AND γ ARE BASICALLY BASED ON [4].)

Emphasis on	Value (a) SDR	Value (b) time
Range $M = N' = K'$	1	4
Number of iteration I	40	5
Weighting constant w	1	1
Weighting constant μ	0.1	0.1
Exponential factor γ	1	1
Frame length of STFT L	1024 (64 [ms])	1024 (64 [ms])
Frame shift of STFT s	256 (16 [ms])	256 (16 [ms])
Window Function	hanning	hanning
Post processing	Wiener mask	Wiener mask

B. Comparative Methods

The methods we compared to were as follows. One was the latest version (release 1.2) of OpenBliSSART [55], [56]. OpenBliSSART is an NMF-based general framework of audio source separation, which is designed with an emphasis on the efficiency of computation for the sake of the practical use. The parameters of OpenBliSSART was chosen on the basis of their recommendations described in the user's manual and demo files. Specifically, the 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 is shown in Table VI.

The other method was median-based harmonic/percussive sound separation proposed by FitzGerald [9], which is another extension of our previous conference paper [4]. The parameters of this method are shown in Table V. The parameter (a) is very similar to the one described in their original paper [9], but not identical. In the original paper [9], the frame length was 92.8 [ms] = 4096/44100 [s], and the frame shift was its quarter. The difference comes from the difference of sampling rate. The parameter (b) is a referential one that may be faster than (a).

TABLE V
PARAMETER SETTINGS OF MEDIAN-FILTERING-BASED METHOD [9].

Parameter	Value (a)	Value (b)
Range of median M	8	4
Frame length of STFT L	1024 (64 [ms])	1024 (64 [ms])
Frame shift of STFT s	256 (16 [ms])	256 (16 [ms])
Window Function	hanning	hanning
Post processing	Wiener mask	Wiener mask

TABLE VI
PARAMETER SETTINGS OF OPENBLISSART [55]. FOR THE DETAILS OF EACH PARAMETER, SEE THE USERS' MANUAL.

Parameter	Value (a)	Value (b)
Number of NMF bases	30	50
Number of NMF Iteration	100 (default)	100
Window size	60 [ms] (default)	60 [ms]
Window overlap	30 [ms] (default)	30 [ms]
Initialization	Gaussian (default)	Gaussian
Window overlap	30 [ms] (default)	30 [ms]
Iteration	100 (default)	100
Epsilon	0 (default)	0
Preenphasis	0 (default)	0
Window function of STFT	sine (default)	sine
Volatile	True (default)	True
Reduce mids	False (default)	False
Remove DC	False (default)	False
Zero-Padding	False (default)	False

C. Dataset and Evaluation Criteria

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

(i) 6 pieces from the MASS database [53]. All of them are monaural, sampled at 16 kHz, and $\approx$ 10 [s] of length.
(ii) 8 of 11 songs from the QUASI (QUaero Audio SIgnals) dataset [57]–[59] 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 the BASS-dB [41] database. 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], 6 of which were used to decide the parameters.

In addition, we applied some simple effectors to the signals, using a software "SoX" (Sound eXchange) [60] version 14.3.2. The effects we applied are following.

Effects	Commands
1) dry (no effect)	(apply nothing)
2) slow (tempo $\times 0.8$)	`sox $in $out tempo 0.8`
3) fast (tempo $\times 1.2$)	`sox $in $out tempo 1.2`
4) overdrive	`sox $in $out overdrive`
5) phaser	`sox $in $out phaser`
6) reverb	`sox $in $out reverb`
7) flanger	`sox $in $out flanger`

We applied each effects to both harmonic and percussive components separately, and mixed them later. Thus we obtained 25×7 clips of length $\approx$ 8–25 [s].

In experiment, we mixed harmonic and percussive components in 5 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, SIR and SAR [54][5], which are commonly used in many source separation tasks. We evaluated the values of both separated $\boldsymbol{H}$ and $\boldsymbol{P}$ components. We considered the processed signals as the "ground truth" instead of the original "dry" ones, when evaluating the performances in the cases of 2)–7).

D. Results and Discussions

Fig. 5 shows SDR, SIR, and SAR of the separated harmonic and percussive signals. In Fig. 5 (a), (b) and (c), it is observed that the HPSS algorithms outperform, or if not, perform almost comparably to the others in terms of SDR and SIR, on average, while Fig. 5 (d) shows that HPSS methods are not as effective as OpenBliSSART in terms of the SIR of percussive components. Fig. 5 (e) and (f) show the SAR of the resultant signals. HPSS 1-A and HPSS 2 (b) outperformed the NMF-based method in terms of SAR of harmonic components, and all the HPSS methods outperformed the NMF-based method in terms of SAR of percussive components, on average.

Fig. 6 shows the results of the cases of processed music signals. In all, it is observed that the shapes of the distributions are not much different from the case of Fig. 5. This fact implies that the performances of the algorithms are rather unaffected by these 6 effects (tempo conversion, phaser, etc.) on average. In other words, the algorithms can pass for many kinds of distorted music signals without changing the parameters drastically, although there may still be a room of tuning the parameters quite elaborately.

Qualitatively, the percussive components output by HPSS methods and the FitzGerald's median often contain many "harmonic" components, too. In other words, these methods "discreetly" erase the harmonic components from $\boldsymbol{P}$. Typical "harmonic" components found in the $\boldsymbol{P}$'s were, the attacks of the piano and the guitar, and the consonants in the singing voice, etc. (Note we discussed in our previous paper [16] that the vocal components (including vowels) are classified into percussive components when L is large, because of the fluctuation of the singing voice.) To the contrary, the percussive components of the OpenBliSSART typically contain few harmonic components, which may be a reason for the rather high SIR.

Generally, the results of the HPSS * (a) sound more clearly than those of the HPSS * (b) for the authors' ears. The results of HPSS 1-A, 1-B, and 2, have minor differences, though it is not necessarily easy to distinguish them by the ears. For the authors' ears, the results of 1-B (a) tend to sound better than the others, though it depends on the cases.

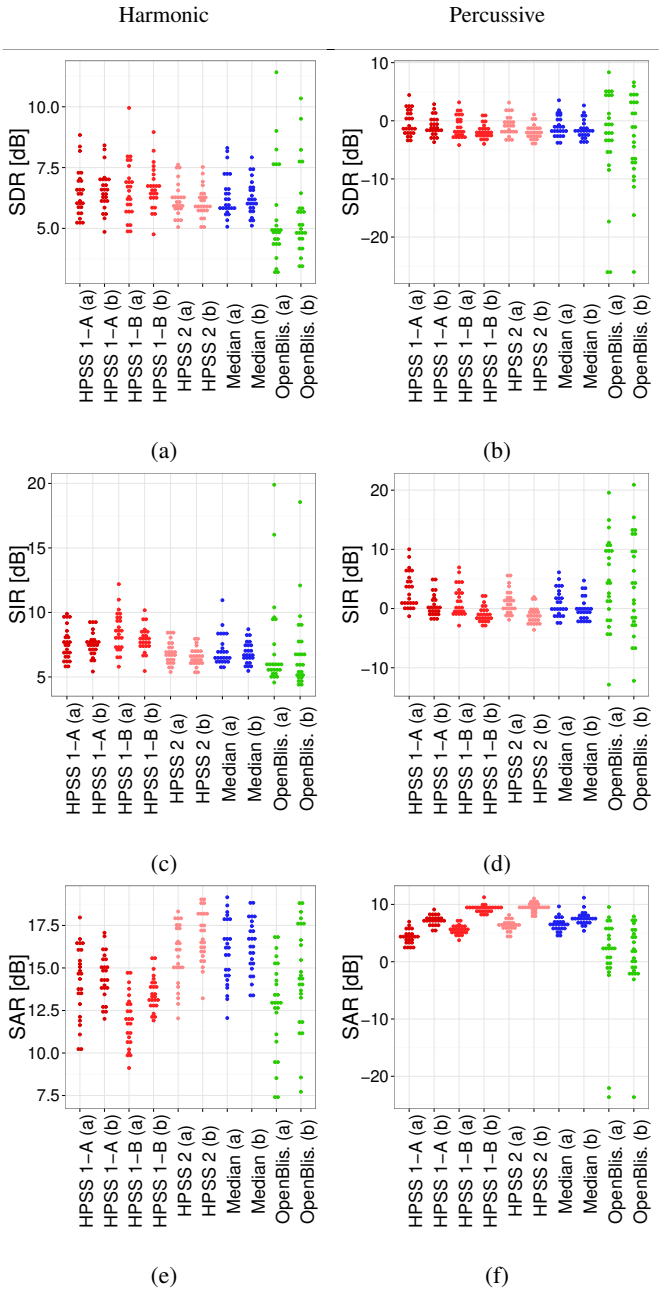

Fig. 5. Dotplot of (a) SDR of harmonic components for 25 songs without effects, (b) SDR of percussive components, (c) SIR of harmonic components, (d) SIR of percussive components, (e) SAR of harmonic components, (f) SAR of percussive components. Each dot indicates a song. "HPSS *" are the proposed methods. "Median" indicates the method proposed by FitzGerald [9]. "OpenBlis." indicates OpenBliSSART [55], [56].

VII. Evaluation of Computation Time

A. Experimental Condition

Practically, the computation cost is an important issue to use a technique in the 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.

In median filtering, we simply applied a naive bin-wise approach; we first applied a sort algorithm around each bin, and picked the center up[6]. The sort algorithm was `std::sort` in C++ standard

[5]We basically ran BSS_EVAL 3.0 [54] (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. However, we deleted the permutation estimation subroutine from the original BSS_EVAL in this experiment, 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.

[6]Robertson et al. [61] proposed a more efficient median filtering algorithm than the naive bin-wise median; the method was approximately 3 times faster than the bin-wise median in their settings. Therefore, the digits in Tables VII, VIII may be multiplied by $1/3$, if the method is used. However, the naive bin-wise median still has an advantage of the parallelism; i.e., it can be achieved by a team of independently-working $N \times K$ agents, each of which is only responsible for the calculation of the median around its own bin. In this case, if there are n cores, the computation time may roughly be $1/n$ times of the digits shown in Table VII, VIII. This is basically the same for the HPSS algorithms. It depends on the computer architecture which algorithm to use. In this paper we used the naive one for the simplicity of the implementation, though $n = 1$ in our experiment.

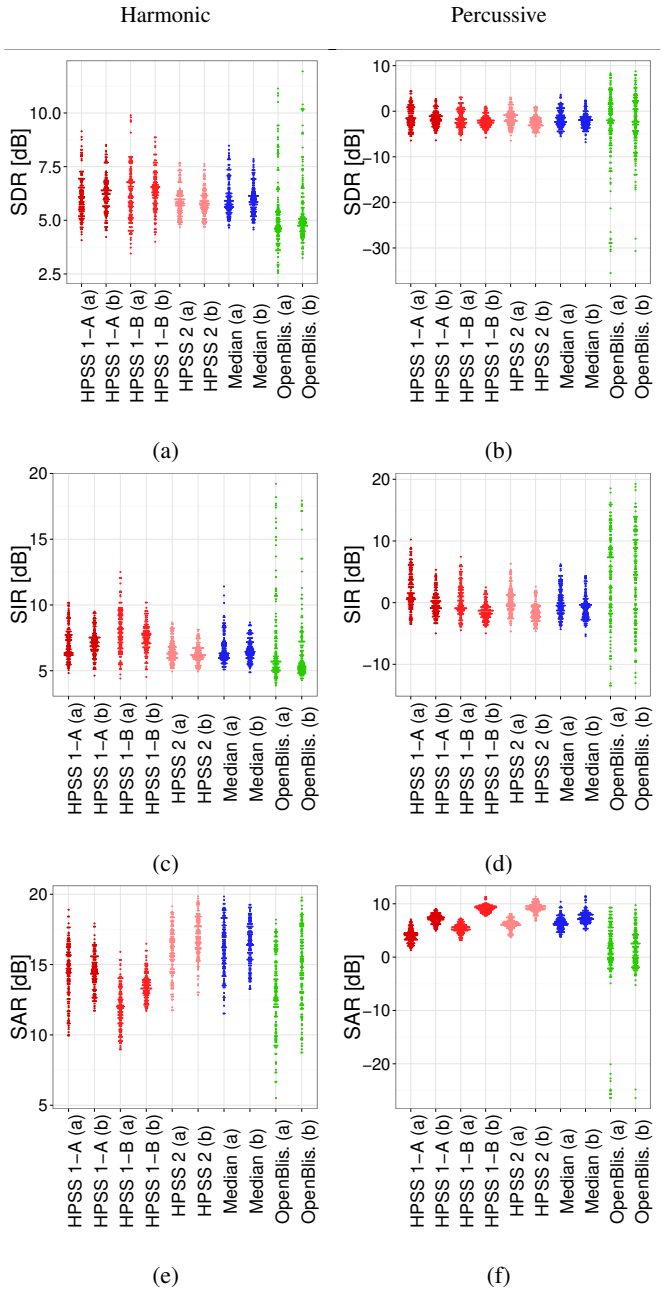

Fig. 6. Dotplot of (a) SDR of harmonic components for 25 songs × 6 effects, (b) SDR of percussive components, (c) SIR of harmonic components, (d) SIR of percussive components, (e) SAR of harmonic components, (f) SAR of percussive components.

library.

All of our methods and FitzGedald's median filtering were implemented in C++ in a same framework by the authors. The type of all the variables were `double` (double floating point number.) In discrete Fourier transform, we used FFTW3 [62], which is one of the standards. We compiled all the programs using the GNU C++ Compiler 4.6.3 (`g++`), with the optimization option `-O3`. The computer we used for evaluation was the same laptop workstation (DELL PRECISION M4500). Although the CPU has 4 cores, we used only a single core for calculation.

B. Results and Discussion

Table VII shows the computation time and the real time factor (RTF). It shows that the proposed methods require quite a short computation time. HPSS 1-B (b) was the fastest of all, and it processed a three-minute song in 4 [s]. These digits indicate that the methods can process the data in real time. (An implementation of real-time HPSS is described in [4]. This is based on performing a sliding analysis on a randomly-accessible queue.)

TABLE VII
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.

Method	60-second song Time	60-second song RTF	180-second song Time	180-second song RTF
HPSS 1-A (a)	2.46 [s]	4.1×10^{-2}	7.5 [s]	4.2×10^{-2}
HPSS 1-A (b)	1.44 [s]	2.4×10^{-2}	4.2 [s]	2.3×10^{-2}
HPSS 1-B (a)	1.77 [s]	3.0×10^{-2}	5.2 [s]	2.9×10^{-2}
HPSS 1-B (b)	1.12 [s]	1.9×10^{-2}	3.9 [s]	2.2×10^{-2}
HPSS 2 (a)	7.45 [s]	12.4×10^{-2}	22.4 [s]	12.4×10^{-2}
HPSS 2 (b)	2.61 [s]	4.4×10^{-2}	7.8 [s]	4.3×10^{-2}
Median (a)	3.48 [s]	5.8×10^{-2}	10.3 [s]	5.7×10^{-2}
Median (b)	2.68 [s]	4.5×10^{-2}	8.7 [s]	4.8×10^{-2}
OpenBlis. (a)	13.0 [s]	22×10^{-2}	37 [s]	21×10^{-2}
OpenBlis. (b)	18.5 [s]	31×10^{-2}	54 [s]	30×10^{-2}

TABLE VIII
COMPUTATION TIME OF EACH METHOD TO PROCESS THE SONGS, *EXCLUDING* SUBSIDIARY PROCESSING SUCH AS I/O. THAT IS, THE DIGITS BELOW ARE ONLY ON EXECUTING THE CORE ALGORITHMS, I.E., UPDATING FORMULAE OF HPSS, APPLYING MEDIAN FILTER, AND NMF. SEE ALSO THE FOOTNOTE 7.

Method	60-second song Time	60-second song RTF	180-second song Time	180-second song RTF
HPSS 1-A (a)	1.28 [s]	2.1×10^{-2}	3.9 [s]	2.1×10^{-2}
HPSS 1-A (b)	.25 [s]	$.42\times10^{-2}$	.67 [s]	$.37\times10^{-2}$
HPSS 1-B (a)	.58 [s]	$.97\times10^{-2}$	1.7 [s]	$.94\times10^{-2}$
HPSS 1-B (b)	.11 [s]	$.18\times10^{-2}$	.36 [s]	$.20\times10^{-2}$
HPSS 2 (a)	5.95 [s]	9.9×10^{-2}	17.7 [s]	9.8×10^{-2}
HPSS 2 (b)	1.10 [s]	1.8×10^{-2}	3.2 [s]	1.8×10^{-2}
Median (a)	1.44 [s]	2.4×10^{-2}	4.3 [s]	2.4×10^{-2}
Median (b)	.61 [s]	1.0×10^{-2}	1.9 [s]	1.1×10^{-2}
OpenBlis. (a)	*11.8* [s][7]		*33* [s][7]	
OpenBlis. (b)	*17.3* [s][7]		*50* [s][7]	

Moreover, considering the computation time of the core of the HPSS, i.e., the HPSS updating formulae, excluding the subsidiary processing such as STFT, Wiener masking, inverse STFT, I/O etc., the HPSS techniques demonstrate outstanding efficiency. It is shown in Table VIII. Comparing the digits shown in Table VII and VIII, we may find that the computation cost of the core HPSS (updating formula) is much less than, or comparable to, the other subsidiary processing. For example, HPSS 1-A (b) requires 4.2 [s] to process a 3-minute song, but only 0.67 [s] of the whole processing time is attributable to the HPSS updating formulae, while the other 3.5 [s] is attributable to the subsidiary processing such as STFT and I/O. This fact also supports the efficiency of HPSS.

When compared to OpenBliSSART, which is a representative of NMF-based methods, the efficiency is especially outstanding. In specific, comparing HPSS 1-B (b) (0.36 [s]) with OpenBliSSART (b) ($\approx$ 50 [s][7]), HPSS may be more than 100 times faster than OpenBliSSART, excluding the subsidiary processing mentioned above. This low computation cost is an advantage of the proposed method compared to other existing methods.

VIII. Final Remarks

A. Summary and Comments

In this paper, we described harmonic/percussive sound separation algorithms, which are based on the anisotropic smoothness of audio spectrograms: that the harmonic component is horizontally smooth and the percussive component is vertically smooth. We showed that the assumption is reasonable using real instrumental sounds, and formulated the problem as an optimization problem to minimize a "smoothness function" which is defined on the basis of the assumption. We specifically formulated the problems in three ways. The difference between them depends on how a requirement for the summation of the separated signals is formulated. Experimental evaluation showed that the performance of these HPSS techniques are higher or comparable to one of the existing methods that is based on NMF, in terms of SxR. In addition, the computation time was much less than it.

Putting all the assumptions in formulation, the derivation procedures and the experimental results together, the characteristics of the proposed three methods may be summarized as follows.

- HPSS 1-A has an advantage that the summation of the separated signals $h(t), p(t)$ is always identical to the input signal when $\gamma = 0.5$, except tiny numerical errors. The form of the updating formula is quite similar to FitzGerald's median filter, which was discussed in the footnote 1.
- HPSS 1-B, especially the parameter set (b), is the fastest of all. It is guaranteed that the updating iteration always converge to a global optimum, but the summation of $h(t)$ and $p(t)$ is not necessarily identical to the input signal because (13) is not necessarily satisfied in the real-world music. According to our experiments, this method with the parameter set (a) may be the most recommendable HPSS, in terms of the trade-off between the SDR and the computation time, though it may not be necessarily always the best.
- HPSS 2 typically requires more computation time than the others, while it is still faster than the NMF (OpenBliSSART). The method achieves a little higher SAR than the others.

B. Future Work

Considering one of the purposes of the HPSS methods to use them as a preprocessor for MIR tasks, an area for future work is the investigation of whether the methods really improve the performance of the applications e.g. chord estimation and tempo estimation, which have been partly verified in e.g. [6], [7]. Future works will include this issue.

Another topic for future is on the choice of time-frequency distribution. This paper considered only the STFT as a time-frequency representation of signals, partly because of its efficiency, as well as the ease of inverse transformation. However, other time-frequency distribution such as the continuous wavelet transform (CWT) and the constant Q transform (CQT) with chromatic frequency resolution, etc., may also be used similarly. In particular, in some MIR applications, the use of a CQT may be more advantageous than an STFT. The use of such time-frequency distributions other than the STFT could be a topic of future work.

With regard to the implementation, despite the rapidity of HPSS, there still is room to accelerate the methods in the hardware level. Since HPSS is based on element-wise simple arithmetic operations, exploiting a parallel architecture e.g. GPU may accelerate computation effectively. These issues regarding implementation would also be challenges in the future.

[7]The computation time of the OpenBliSSART excluding I/O, STFT, etc., shown in Table VIII, were not directly measured but estimated just by subtracting 1.2 [s] (60-second song) and 3.5 [s] (3-minute song) from the total computation time shown in Table VII, just for reference. The values 1.2 [s] and 3.5 [s] above were estimated by the cases of HPSS, since the computation time of I/O, STFT, etc., would be a constant regardless of the source separation algorithms. The reason the authors did not measure the core computation time of OpenBliSSART was that the authors did not have enough knowledge on the details of the implementation of the OpenBliSSART, which was required to embed a timer inside of the source codes.

Acknowledgement

The authors would like to thank the anonymous reviewers for their constructive suggestions.

References

[1] 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," in *Proc. Acoustical Society of Japan Autumn Conference (in Japanese)*, 2007, pp. 825–826.

[2] K. Miyamoto, H. Kameoka, N. Ono, and S. Sagayama, "Separation of harmonic and non-harmonic sounds based on anisotropy in spectrogram," in *Proc. Acoustical Society of Japan Spring Conference (in Japanese)*, 2008, pp. 903–904.

[3] 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 *Proc. EUSIPCO*, 2008.

[4] N. Ono, K. Miyamoto, H. Kameoka, and S. Sagayama, "A real-time equalizer of harmonic and percussive components in music signals," in *Proc. ISMIR*, 2008, pp. 139–144.

[5] H. Tachibana, H. Kameoka, N. Ono, and S. Sagayama, "Comparative evaluation of multiple harmonic/percussive sound separation techniques based on anisotropic smoothness of spectrogram," in *Proc. ICASSP*, 2012, pp. 465–468.

[6] J. Reed, Y. Ueda, S. M. Siniscalchi, Y. Uchiyama, and S. Sagayama, "Minimum classification error training to improve isolated chord recognition," in *proc. ICASSP*, 2009, pp. 609–614.

[7] A. Gkiokas, V. Katsouros, G. Carayannis, and T. Stafylakis, "Music tempo estimation and beat tracking by applying source separation and metrical relations," in *Proc. ICASSP*, 2012, pp. 421–424.

[8] N. Q. K. Duong, H. Tachibana, E. Vincent, N. Ono, R. Gribonval, and S. Sagayama, "Multichannel harmonic and percussive component separation by joint modeling of spatial and spectral continuity," in *Proc. ICASSP*, 2011, pp. 205–208.

[9] D. FitzGerald, "Harmonic/percussive separation using median filtering," in *Proc. DAFx*, 2010.

[10] 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.

[11] Y. Ueda, Y. Uchiyama, T. Nishimoto, N. Ono, and S. Sagayama, "HMM-based approach for automatic chord detection using refined acoustic features," in *proc. ICASSP*, 2010, pp. 5518–5521.

[12] E. Tsunoo, N. Ono, and S. Sagayama, "Rhythm map: Extraction of unit rhythmic patterns and analysis of rhythmic structure from music acoustic signals," in *Proc. ICASSP*, 2009, pp. 185–188.

[13] 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.

[14] H. Tachibana, T. Ono, N. Ono, and S. Sagayama, "Melody line estimation in homophonic music audio signals based on temporal-variability of melodic source," in *Proc. ICASSP*, 2010, pp. 425–428.

[15] C.-L. Hsu, D. L. Wann, and J.-S. R. Jang, "A trend estimation algorithm for singing pitch detection in musical recordings," in *Proc. ICASSP*, 2011, pp. 393–396.

[16] H. Tachibana, N. Ono, and S. Sagayama, "Singing voice enhancement in monaural music signals based on two-stage harmonic/percussive sound separation on multiple resolution spectrograms," *IEEE/ACM Trans. on Audio, Speech, and Lang. Process.*, vol. 22, no. 1, pp. 228–237, 2014.

[17] 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.
[18] C. Duxbury, M. Davies, and M. Sandler, "Separation of transient information in music audio using multiresolution analysis techniques," in *Proc. DAFX01*, 2001.
[19] A. Zils, F. Pachet, O. Delerue, and F. Gouyon, "Automatic extraction of drum tracks from polyphonic music signals," in *Proc. of the 2nd International Conference on Web Delivering of Music*, 2002.
[20] K. Yoshii, M. Goto, and H. G. Okuno, "Automatic drum sound description for real-world music using template adaptation and matching methods," in *Proc. ISMIR*, 2004, pp. 184–191.
[21] 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.
[22] D. Barry, D. FitzGerald, and E. Coyle, "Drum source separation using percussive feature detection and spectral modulation," in *Proc. IEE Irich Signals and Systems Conference*, 2005.
[23] Z. Duan, Y. Zhang, C. Zhang, and Z. Shi, "Unsupervised single-channel music source separation by average harmonic structure modeling," *IEEE Trans. Audio, Speech, and Sig. Process.*, vol. 16, no. 4, pp. 766–778, 2008.
[24] O. Gillet and G. Richard, "Transcription and separation of drum signals from polyphonic music," *IEEE Trans. on Audio, Speech, and Sig. Proc.*, vol. 16, no. 3, pp. 529–540, 2008.
[25] F. Rigaud, M. Lagrange, A. Röbel, and G. Peeters, "Drum extraction from polyphonic music based on a spectro-temporal model of percussive sounds," in *Proc. ICASSP*, 2011, pp. 381–384.
[26] W. Sethares, "Local consonance and the relation ship between timbre and scale," *The Journal of Acoustical Society of America*, vol. 94, no. 3, pp. 1218–, 1993.
[27] ISO, "Information technology – multimedia content description interface – part 4: Audio, ISO-IEC 15938-4 (E)," 2001.
[28] A. Hyvärinen, J. Karhunen, and E. Oja, *Independent Component Analysis*. New York: Wiley, 2001.
[29] D. D. Lee and H. S. Seung, "Algorithms for non-negative matrix factorization," in *Proc. NIPS*, 2000, pp. 556–562.
[30] P. Smaragdis and J. C. Brown, "Non-negative matrix factorization for polyphonic music transcription," in *Proc. WASPAA*, 2003, pp. 177–180.
[31] M. Helen and T. Virtanen, "Separation of drums from polyphonic music using non-negative matrix factorization and support vector machine," in *Proc. EUSIPCO*, 2005.
[32] T. Heittola and A. Klapuri, "Locating segments with drums in music signals," in *in Proc. ISMIR*, 2002.
[33] J. Paulus and T. Virtanen, "Drum transcription with non-negative spectrogram factorization," in *Proc. EUSIPCO*, 2005.
[34] A. Moreau and A. Flexer, "Drum transcription in polyphonic music using non-negative matrix factorization," in *Proc. ISMIR*, 2007.
[35] 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.
[36] J. Yoo, M. Kim, K. Kang, and S. Choi, "Nonnegative matrix partial cofactorization for drum source separation," in *Proc. ICASSP*, 2010, pp. 1942–1945.
[37] M. Kim, J. Yoo, K. Kang, and S. Choi, "Nonnegative matrix partial cofactorization for spectral and temporal drum source separation," *IEEE Journal of Selected Topics in Signal Processing*, vol. 5, no. 6, p. 1192, 2011.
[38] ——, "Blind rhythmic source separation: nonnegativity and repeatability," 2010.
[39] T. Virtanen, "Monaural sound source separation by nonnegative matrix factorization with temporal continuity and sparseness criteria," *IEEE Trans. on Audio, Speech and Lang. Proc.*, vol. 3, no. 15, pp. 1066–1074, 2007.
[40] E. Vincent, N. Berlin, and R. Badeau, "Harmonic and in-harmonic nonnegative matrix factorization for polyphonic pitch transcription," in *Proc. ICASSP*, 2008.
[41] E. Vincent, R. Gribonval, and C. Févotte, "BASS-dB: the blind audio source separation evaluation database." [Online]. Available: http://www.inria.fr/metiss/BASS-dB/
[42] T. Virtanen, A. T. Cemgil, and S. J. Godsill, "Bayesian extensions to nonnegative matrix factorization for audio signal processing," in *Proc. ICASSP*, 2008.
[43] O. Dikmen and T. Cemgil, "Unsupervised single-channel source separation using bayesian NMF," in *Proc. WASPAA*, 2009, pp. 93–96.
[44] A. T. Cemgil and O. Dikmen, "Conjugate gamma markov random field for modelling nonstationary sources," in *Proc. ICA*, 2007, pp. 697–705.
[45] D. FitzGerald, E. Coyle, and M. Cranitch, "Using tensor factorisation models to separate drums from polyphonic music," in *Proc. DAFx*, 2009.
[46] T. Virtanenm, A. Mesaros, and M. Ryynänen, "Combining pitch-based inference and non-negative spectrogram factorization in separating vocals from polyphonic music," *in Proc SAPA*, pp. 17–20, 2008.
[47] T. Virtanen, "Sound source separation using sparse coding with temporal continuity objectives," in *Proc. ICMC*, 2003, pp. 231–234.
[48] S. S. Stevens, "The measurement of loudness," *Journal of Acoustical Society of America*, vol. 27, pp. 815–829, 1955.
[49] M. Goto, "Development of the RWC music database," *Proceedings of the 18th International Congress on Acoustics (ICA 2004)*, pp. I–553–556, 2004.
[50] 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.
[51] H. Karcher, "Riemann center of mass and mollifier smoothing," *Communications on Pure and Applied Mathematics*, vol. 30, pp. 509–541, 1977.
[52] D. R. Hunter and K. Lange, "Quantile regression via an MM algorithm," *Journal of Computational and Graphical Statistics*, vol. 9, pp. 60–77, 2000.
[53] M. Vinyes, "MTG MASS database," 2008, http://www.mtg.upf.edu/static/mass/resources.
[54] E. Vincent, R. Gribonval, and C. Févotte, "Performance measurement in blind audio source separation," vol. 14, no. 4, pp. 1462–1469, 2006.
[55] F. Weninger, A. Lehmann, and B. Schuller, "OpenBLiSSART: Design and evaluation of a research toolkit for blind source separation in audio recognition tasks," in *in Proc. ICASSP*, 2011, pp. 1625–1628.
[56] [Online]. Available: http://openblissart.github.com/OpenBliSSART/
[57] 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.
[58] A. Liutkus, R. Badeau, and G. Richard, "Gaussian process for underdetermined source separation," *IEEE Trans. Sig. Proc.*, vol. 50, no. 7, pp. 3155–3167, 2011.
[59] "QUASI database – a musical audio signal database for source separation." [Online]. Available: http://www.tsi.telecom-paristech.fr/aao/en/2012/03/12/quasi/
[60] [Online]. Available: http://sox.sourceforge.net
[61] A. Robertson, A. Stark, and M. E. P. Davis, "Percussive beat tracking using real-time median filtering," in *Proc. International Workshop on Machine Learning and Music ECML/PKDD*, 2013.
[62] M. Frigo and S. G. Johhanson, "The design and implementation of FFTW3," *Proceedings of IEEE*, vol. 93, no. 2, 2005.

Hideyuki Tachibana (S'10) received a B.E. degree in mathematical engineering and information physics and an M.E. and Ph.D degrees in information physics and computing from the University of Tokyo, Tokyo, Japan, in 2008, 2010, and 2014, respectively. His research interests include signal processing and artificial intelligence on music. He is a member of IEEE, ACM, Acoustical Society of Japan (ASJ), Information Processing Society of Japan (IPSJ), and the Institute of Electronics, Information and Communication Engineers (IEICE).

Nobutaka Ono (M '02) received the B.E., M.S., and Ph.D degrees in Mathematical Engineering and Information Physics from the University of Tokyo, Japan, in 1996, 1998, 2001, respectively. He joined the Graduate School of Information Science and Technology, the University of Tokyo, Japan, in 2001 as a Research Associate and became a Lecturer in 2005. He moved to the National Institute of Informatics, Japan, as an Associate Professor in 2011. His research interests include acoustic signal processing, specifically, microphone array processing, source localization and separation, music signal processing, audio coding and watermarking, and optimization algorithms for them. He is the author or co-author of more than 100 articles in international journal papers and peer-reviewed conference proceedings. Dr. Ono has been an Associate Editor of the IEEE Transactions on Audio, Speech and Language Processing since 2012 and also an Associate Editor of Acoustic Society and Technology since 2012. He was a Tutorial speaker at ISMIR 2010 and organized a special session in EUSIPCO 2013. He is a chair of SiSEC (Signal Separation Evaluation Campaign) evaluation committee in 2013. He is a Senior member of the IEEE Signal Processing Society, and a member of the Institute of Electronics, Information and Communications Engineers (IEICE), the Acoustical Society of Japan (ASJ), the Information Processing Society of Japan (IPSJ), the Institute of Electrical Engineers of Japan (IEEJ), and the Society of Instrument and Control Engineers (SICE). He received the Sato Paper Award and the Awaya Award from ASJ in 2000 and 2007, respectively, received the Igarashi Award at the Sensor Symposium on Sensors, Micromachines, and Applied Systems from IEEJ in 2004, and received the best paper award in IEEE International Symposium on Industrial Electronics (ISIE) in 2008.

Shigeki Sagayama (M' 82) received the B.E., M.S., and Ph.D. degrees from the University of Tokyo, Tokyo, Japan, in 1972, 1974, and 1998, respectively, all in mathematical engineering and information physics. He joined Nippon Telegraph and Telephone Public Corporation (currently, NTT) in 1974 and started his career in speech analysis, synthesis, and recognition at NTT Labs in Musashino, Japan. From 1990, he was Head of the Speech Processing Department, ATR Interpreting Telephony Laboratories, Kyoto, Japan where he was in charge of an automatic speech translation project. In 1993, he was responsible for speech recognition, synthesis, and dialog systems at NTT Human Interface Laboratories, Yokosuka, Japan. In 1998, he became a Professor of the Graduate School of Information Science, Japan Advanced Institute of Science and Technology (JAIST), Ishikawa. In 2000, he was appointed Professor at the Graduate School of Information Science and Technology (formerly, Graduate School of Engineering), the University of Tokyo. On his retirement from the University of Tokyo in 2013, he became a Project Professor at the National Institute of Informatics (NII). His major research interests include the processing and recognition of speech, music, acoustic signals, handwriting, and images. He was the leader of anthropomorphic spoken dialog agent project (Galatea Project) from 2000 to 2003. Prof. Sagayama received the National Invention Award from the Institute of Invention of Japan in 1991, the Chief Official's Award for Research Achievement from the Science and Technology Agency of Japan in 1996, and other academic awards including Paper Awards from the Institute of Electronics, Information and Communications Engineers, Japan (IEICEJ) in 1996 and from the Information Processing Society of Japan (IPSJ) in 1995. He is a member of the Acoustical Society of Japan, IEICEJ, and IPSJ.

Hirokazu Kameoka (M'07) received B.E., M.S., and Ph.D. degrees all from the University of Tokyo, Tokyo, Japan, in 2002, 2004, and 2007 respectively. He is currently a Research Scientist at NTT Communication Science Laboratories and a Visiting Associate Professor at the University of Tokyo. His research interests include computational auditory scene analysis, statistical signal processing, speech and music processing, and machine learning. He is a member of IEEE, the Information Processing Society of Japan (IPSJ), and the Acoustical Socket of Japan (ASJ). He received 13 awards over the past 10 years, including the Yamashita Memorial Research Award in 2005 from IPSJ, the Itakura Prize Innovative Young Research Award in 2007 and the Awaya Prize Young Researcher Award in 2008 from ASJ, IEEE Signal Processing Society 2008 SPS Young Author Best Paper Award in 2009 and IEEE ISS Young Researcher's Award in Speech Field in 2011.