Available online at www.sciencedirect.com

ScienceDirect

Signal Processing 87 (2007) 1833–1847

SIGNAL PROCESSING

www.elsevier.com/locate/sigpro

Underdetermined blind sparse source separation for arbitrarily arranged multiple sensors

Shoko Araki[a,b,*], Hiroshi Sawada[a], Ryo Mukai[a], Shoji Makino[a,b]

[a]*NTT Communication Science Laboratories, NTT Corporation, 2–4 Hikaridai, Seika-cho, Soraku-gun, Kyoto 619-0237, Japan*
[b]*Graduate School of Information Science and Technology, Hokkaido University, Kita 14, Nishi 9,*
Kita-ku, Sapporo-shi, Hokkaido 060-0814, Japan

Received 31 July 2006; received in revised form 7 February 2007; accepted 9 February 2007
Available online 1 March 2007

Abstract

This paper presents a new method for blind sparse source separation. Some sparse source separation methods, which rely on source sparseness and an anechoic mixing model, have already been proposed. These methods utilize level ratios and phase differences between sensor observations as their features, and they separate signals by classifying them. However, some of the features cannot form clusters with a well-known clustering algorithm, e.g., the *k*-means. Moreover, most previous methods utilize a linear sensor array (or only two sensors), and therefore they cannot separate symmetrically positioned sources. To overcome such problems, we propose a new feature that can be clustered by the *k*-means algorithm and that can be easily applied to more than three sensors arranged non-linearly. We have obtained promising results for two- and three-dimensionally distributed speech separation with non-linear/non-uniform sensor arrays in a real room even in underdetermined situations. We also investigate the way in which the performance of such methods is affected by room reverberation, which may cause the sparseness and anechoic assumptions to collapse.

Keywords: Blind source separation; Sparseness; Clustering; Normalization; Binary mask; Speech separation; Reverberation

1. Introduction

Blind source separation (BSS) [1] is an approach for estimating source signals that uses only the mixed signal information observed at each sensor. The BSS technique for speech dealt with in this paper has many applications including hands-free teleconference systems and automatic conference minute generators.

Two approaches have been widely studied and employed to solve the BSS problem; one is based on independent component analysis (ICA) (e.g., [2]) and the other relies on the sparseness of source signals (e.g., [3]). Recently, many ICA methods have been proposed even for the convolutive BSS problem [2,4–10]. ICA works well even in a reverberant condition when the number of sources N is less than or equal to the number of sensors M. On the other hand, the sparseness-based approaches are attractive because they can handle the underdetermined problem, i.e., $N>M$.

The sparseness-based approaches can be divided into two main categories. One method is based on

*Corresponding author. NTT Communication Science Laboratories, NTT Corporation, 2–4 Hikaridai, Seika-cho, Soraku-gun, Kyoto 619-0237, Japan. Tel.: +81 774 93 5319; fax: +81 774 93 5158.
E-mail address: shoko@cslab.kecl.ntt.co.jp (S. Araki).

0165-1684/$ - see front matter © 2007 Elsevier B.V. All rights reserved.
doi:10.1016/j.sigpro.2007.02.003

MAP estimation, where the sources are estimated after mixing matrix estimation [11–17], and the other extracts each signals with time-frequency binary masks [3,18–20]. The former method includes mixing matrix estimation and l_1-norm minimization in the frequency domain (i.e., for complex numbers), both of which still present difficulties [16]. The latter, the binary mask approach, has the advantage of being implemented in real time [21]. In this paper we focus on the binary mask approach.

In the binary mask approach, we assume that signals are sufficiently sparse, and therefore, we can assume that at most one source is dominant at each time–frequency slot. If this assumption holds, a histogram of the level and frequency normalized phase differences between two sensor observations has N clusters [3,18,20]. Because an individual cluster in the histogram corresponds to an individual source, we can separate each signal by selecting the observation signal at time–frequency points in each cluster with a binary mask. The best-known approach may be the Degenerate Unmixing Estimation Technique (DUET) [3,18,21].

Previously, such clustering was performed manually [3,18], by using kernel density estimation [20], or with an ML-based gradient method [21]. On the other hand, if clustering could be performed with a well-known algorithm such as the k-means clustering or hierarchical clustering [22], the clustering will be automated and simplified. To employ a widely utilized clustering algorithm such as the k-means, we should be careful about the variances of multiple variables, in this case the level ratios and phase differences. However, frequency normalization of the phase difference, which is important in terms of avoiding the permutation problem among frequencies [16,17], sometimes makes the phase difference much smaller than the level ratio as shown in Section 3.2. Such different variances between the features make clustering with the k-means difficult. This is the prime motivation for this work.

Our second motive is to employ more than three sensors arranged two- or three-dimensionally, which could have a non-linear/non-uniform alignment. Only a few authors have generalized [16,17,23] a method for more than two sensors. Authors of [23] used up to eight sensors, however, their sensors were still linearly arranged. The paper [24] has already tried a multichannel DUET (DESPRIT) by combining the sparse assumption and the Estimation of Signal Parameters via Rotational Invariance Technique (ESPRIT); however, their method still limits the array shape: a linear array or two sets of congruent arrays. A two-sensor system and a linear sensor array limits the separation ability on a two-dimensional half-plane, e.g., the previous methods cannot separate sources placed in a mirror image arrangement. To allow the free location of sources, we need more than three sensors arranged two- or three-dimensionally.

Based on these two motivations, we propose a new binary mask approach MENUET (Multiple sENsor dUET), which employs the well-known k-means clustering algorithm. As a feature, our method utilizes the level ratios and phase differences between multiple observations. To realize level ratio and phase difference variances of a comparable level, we propose a way of weighting the phase term for successful clustering. Moreover, our proposed method does not require sensor location information. This allows us to employ freely arranged multiple sensors easily. Therefore, the proposed method can separate signals that are distributed two- or three-dimensionally. Our previous paper, [16], utilized a two-dimensional sensor array to test the MAP approach proposed in [16]. However, that work did not employ the frequency normalization, and therefore, suffered from the abovementioned permutation problem. On the other hand, in this paper, we employ appropriate frequency normalization for the k-means algorithm. Moreover, we also apply our proposed method to a three-dimensional sensor array, and describe the result.

An additional contribution of this paper is that it undertakes an investigation of the separation performance in real world acoustic environments. Both our proposed method and previous methods employ assumptions of source sparseness and anechoic mixing (i.e., a simple attenuation and delay model for a room impulse response). Such assumptions can easily be affected by reverberation. We show how the performance is affected when the problem does not satisfy the assumed conditions.

The organization of this paper is as follows. Section 2 presents the basic framework of the binary mask-based BSS method. In Section 3, we describe some features for clustering, and test how each feature will be clustered by the k-means clustering algorithm. In Section 4, we propose a novel method MENUET, which includes the estimation of geometric features from multiple sensor observations. Our proposed feature is suitable for k-means clustering. Section 5 reports some experimental results obtained with non-linearly arranged sensors in underdetermined scenarios. Even when the sources and sensors were distributed two- or three-dimensionally, we obtained good separation results with the k-means algorithm for each scenario under weak reverberant ($\mathrm{RT}_{60} = 128\,\mathrm{ms}$) conditions. We also investigated the performance under more reverberant conditions ($\mathrm{RT}_{60} = 300\,\mathrm{ms}$). The final section concludes this paper.

2. Binary mask approach to BSS

2.1. Problem description

Suppose that sources $\mathrm{s}_1, \ldots, \mathrm{s}_N$ are convolutively mixed and observed at M sensors

$$\mathrm{x}_j(t) = \sum_{k=1}^{N} \sum_{l} \mathrm{h}_{jk}(l)\mathrm{s}_k(t-l), \quad j = 1, \ldots, M, \tag{1}$$

where $\mathrm{h}_{jk}(l)$ represents the impulse response from source k to sensor j. In this paper, we focus particularly on a situation where the number of sources N can exceed the number of sensors M ($N > M$). We assume that N and M are known, and that the sensor spacing is small enough to avoid the spatial aliasing problem. The goal is to obtain separated signals $\mathrm{y}_k(t)$ that are estimations of s_k solely from M observations.

2.2. Separation procedures

Step 1. Signal transformation to the time–frequency domain: Fig. 1 shows the flow of the binary mask approach. The binary mask approach usually employs a time–frequency domain representation. First, time–domain signals $\mathrm{x}_j(t)$ sampled at frequency f_s are converted into frequency–domain time-series signals $\mathrm{x}_j(f, t)$ with a T-point short-time Fourier transform (STFT):

$$x_j(f,t) \leftarrow \sum_{r=-T/2}^{T/2-1} \mathrm{x}_j(r + tS)\,\mathrm{win}(r)\mathrm{e}^{-j2\pi fr}, \tag{2}$$

where $f \in \{0, (1/T)f_s, \ldots, ((T-1)/T)f_s\}$ is a frequency, $\mathrm{win}(r)$ is a window that tapers smoothly to zero at each end, t is a new index representing time, and S is the window shift size. As the window $\mathrm{win}(r)$, in this paper, we utilized a Hanning window $\frac{1}{2}(1 - \cos(2\pi r/T))$ $(r = 0, \ldots, T-1)$.

There are two advantages to working in the time–frequency domain. First, convolutive mixtures (1) can be approximated as instantaneous mixtures at each frequency:

$$x_j(f,t) \approx \sum_{k=1}^{N} h_{jk}(f) s_k(f,t) \tag{3}$$

or in a vector notation,

$$\mathbf{x}(f,t) \approx \sum_{k=1}^{N} \mathbf{h}_k(f) s_k(f,t), \tag{4}$$

where $h_{jk}(f)$ is the frequency response from source k to sensor j, and $s_k(f,t)$ is a frequency–domain time-series signal of $\mathrm{s}_k(t)$ obtained by the same operation as (2), $\mathbf{x} = [x_1, \ldots, x_M]^{\mathrm{T}}$, and $\mathbf{h}_k = [h_{1k}, \ldots, h_{Mk}]^{\mathrm{T}}$ is a mixing vector that consists of the frequency responses from source s_k to all sensors. The second advantage is that the sparseness of a source signal becomes prominent in the time–frequency domain [12,19], if the source is colored and non-stationary such as speech. The possibility of $s_k(f,t)$ being close to zero is much higher than that of $\mathrm{s}_k(t)$. When the signals are sufficiently sparse in the time–frequency domain, we can assume that the sources rarely overlap and, (3) and (4), respectively, can be approximated as

$$x_j(f,t) \approx h_{jk}(f) s_k(f,t), \quad {}^{\exists}k \in \{1, \ldots, N\}, \tag{5}$$

$$\mathbf{x}(f,t) \approx \mathbf{h}_k(f) s_k(f,t), \quad {}^{\exists}k \in \{1, \ldots, N\}, \tag{6}$$

where $s_k(f,t)$ is a dominant source at the time–frequency point (f,t). For instance this is approximately true for speech signals [3,15]. Fig. 2(a) shows example spectra of three speech sources, in which we can see their temporal/frequency sparseness.

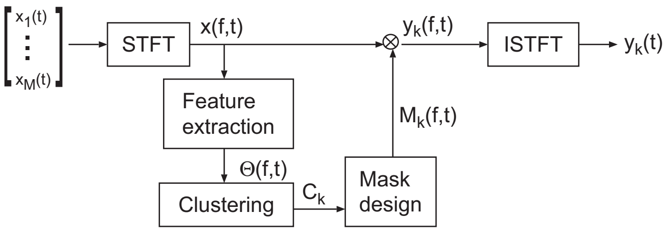

Fig. 1. Basic scheme of binary mask approach.

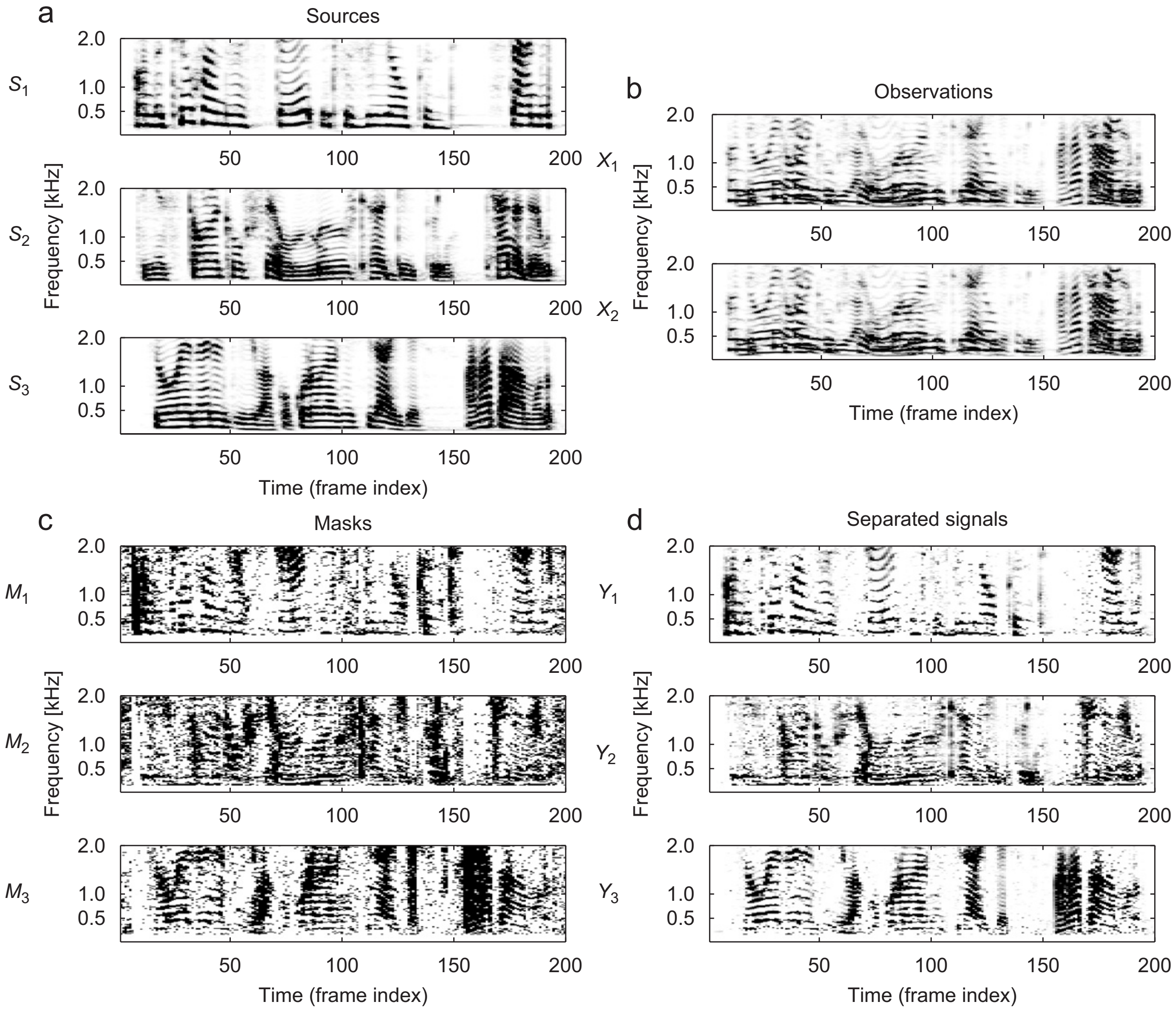

Fig. 2. Example spectra of (a) speech sources, (b) observations, (c) masks and (d) separated signals ($N = 3$, $M = 2$).

Step 2. *Feature extraction*: If the sources $s_k(f,t)$ are sufficiently sparse, separation can be realized by gathering the time–frequency points (f,t) where only one signal s_k is estimated to be dominant. To estimate such time–frequency points, some features $\mathbf{\Theta}(f,t)$ are calculated by using the frequency–domain observation signals $\mathbf{x}(f,t)$. Here, $\mathbf{\Theta}(f,t)$ is a vector that consists of certain geometric features. Generally, the level ratios and phase differences between observations are utilized for $\mathbf{\Theta}(f,t)$. Previously employed features are discussed in Section 3, and our newly proposed feature is introduced in Section 4.

Step 3. *Clustering*: Then the features $\mathbf{\Theta}(f,t)$ are grouped into N clusters $C_1,\ldots,C_N$, where N is the number of possible sources. Formerly, such clustering was undertaken manually [3,18], with a kernel density estimation [20] or with an ML-based gradient method [21]. On the other hand, if we can employ a standard clustering algorithm such as the k-means algorithm or hierarchical clustering [22], the clustering procedure will be automated and simplified. In this work we utilize the k-means clustering algorithm [22] with a given source number N. The clustering criterion is to minimize the total sum $\mathscr{J}$ of the Euclidean distances (ED) between cluster members and their centroids $\mathbf{c}_k$:

$$\mathscr{J} = \sum_{k=1}^{M} \mathscr{J}_k, \quad \mathscr{J}_k = \sum_{\mathbf{\Theta}(f,t)\in C_k} \|\mathbf{\Theta}(f,t) - \mathbf{c}_k\|^2. \tag{7}$$

After setting appropriate initial centroids $\mathbf{c}_k$ ($k = 1, \ldots, N$), this $\mathcal{J}$ can be minimized by the following iterative updates:

$$C_k = \{\boldsymbol{\Theta}(f,t) \mid k = \underset{k}{\operatorname{argmin}} \|\boldsymbol{\Theta}(f,t) - \mathbf{c}_k\|^2\}, \tag{8}$$

$$\mathbf{c}_k \leftarrow E[\boldsymbol{\Theta}(f,t)]_{\boldsymbol{\Theta}\in C_k}, \tag{9}$$

where $E[\cdot]_{\boldsymbol{\Theta}\in C_k}$ is a mean operator for the members of a cluster C_k. The cluster members are determined by (8). If the feature $\boldsymbol{\Theta}(f,t)$ is properly chosen, then each cluster corresponds to an individual source.

Here, it should be noted that the k-means clustering utilizes the ED $\|\boldsymbol{\Theta}(f,t) - \mathbf{c}_k\|^2$, not the Mahalanobis distance (MD) $(\boldsymbol{\Theta}(f,t) - \mathbf{c}_k)^{\mathrm{T}}\boldsymbol{\Sigma}_k^{-1}(\boldsymbol{\Theta}(f,t) - \mathbf{c}_k)$, where $\boldsymbol{\Sigma}_k$ is the covariance matrix of cluster k. That is, k-means assumes clusters of a multivariate isotropic variance $\boldsymbol{\Sigma}_k = \boldsymbol{I}$ for all k, where $\boldsymbol{I}$ denotes an identity matrix.

Step 4. *Mask design*: Next, the separated signals $y_k(f,t)$ are estimated based on the clustering result. We design a time–frequency domain binary mask that extracts the time–frequency points of each cluster

$$M_k(f,t) = \begin{cases} 1, & \boldsymbol{\Theta}(f,t) \in C_k, \\ 0 & \text{otherwise}. \end{cases} \tag{10}$$

Example of binary mask spectra is shown in Fig. 2(c). Then, applying the binary masks (Fig. 2(c)) to one of the observations (Fig. 2(b)) $x_J(f,t)$, we obtain separated signals (Fig. 2(d)):

$$y_k(f,t) = M_k(f,t)x_J(f,t),$$

where J is a selected sensor index.

Step 5. *Separated signal reconstruction*: At the end of the flow (Fig. 1), we obtain outputs $\mathrm{y}_k(t)$ by employing an inverse STFT (ISTFT) and the overlap-and-add method [25],

$$\mathrm{y}_k(t) = \frac{1}{A}\sum_{l=0}^{S-1} \mathrm{y}_k^{m+l}(t), \tag{11}$$

where $A = \frac{1}{2}T/S$ is a constant for the Hanning window case,

$$\mathrm{y}_k^m(t) = \begin{cases} \sum_{f\in\{0,\frac{1}{T}f_s,\ldots,\frac{T-1}{T}f_s\}} y(f,m)\mathrm{e}^{\jmath 2\pi f r}, \\ \quad (mS \leqslant t \leqslant mS + T - 1), \\ 0 \quad (\text{otherwise}), \end{cases}$$

and $r = t - mS$.

3. Discussion of features

Because the binary mask approach depends strongly on the clustering of the feature vectors $\boldsymbol{\Theta}(f,t)$, the selection of an appropriate feature vector $\boldsymbol{\Theta}(f,t)$ is essential to this approach. In this section, we provide examples of the features $\boldsymbol{\Theta}(f,t)$ including the previously utilized feature. We also test how each feature will be clustered by the k-means algorithm.

3.1. Features

Most previous methods utilized the level ratio and/or phase difference between observations as their features $\boldsymbol{\Theta}(f,t)$. The previously proposed features can be summarized as

$$\boldsymbol{\Theta}(f,t) = \left[\frac{|x_2(f,t)|}{|x_1(f,t)|}, \arg\left[\frac{x_2(f,t)}{x_1(f,t)}\right]\right]^{\mathrm{T}} \tag{12}$$

and some examples are shown in Table 1.

Such features (12) represent geometric information on sources and sensors, if the sources are sufficiently sparse. Let us assume that the mixing process is expressed by

$$h_{jk}(f) \approx \lambda_{jk} \exp[-\jmath 2\pi f \tau_{jk}], \tag{13}$$

where $\lambda_{jk} \geqslant 0$ and τ_{jk} are the attenuation and the time delay from source k to sensor j. If there is no reverberation (i.e., an anechoic situation), λ_{jk} and τ_{jk} are determined solely by the geometric distribution of the sources and sensors. If the sources are sparse (5), the feature vector (12) becomes

$$\boldsymbol{\Theta}(f,t) = \left[\frac{\lambda_{2k}}{\lambda_{1k}}, -2\pi f(\tau_{2k} - \tau_{1k})\right]^{\mathrm{T}}, \quad {}^{\exists}k. \tag{14}$$

We can see that $\boldsymbol{\Theta}(f,t)$ contains geometric information on the dominant source s_k at each time-frequency point (f,t).

To avoid frequency dependence in the phase difference (14), some authors have employed a frequency normalization that involves dividing the phase difference by $2\pi f$ or $2\pi f c^{-1} d$ where c is the propagation velocity and d is the sensor spacing (see Table 1). The former is utilized in [3,18] and the latter gives the directions of arrival (DOA) of sources if the sensor spacing d is given correctly [26]. If we do not use such frequency normalization, we have to solve the permutation problem among frequencies after clustering the features [16,17]. Moreover, frequency normalization makes it

Table 1
Typical features and their separation performance (SIR improvement in dB) with the *k*-means algorithm

	Feature $\mathbf{\Theta}(f,t)$		*k*-means	Opt.(ED)	Opt. (MD)
(A)	$\left[\frac{\lvert x_2(f,t)\rvert}{\lvert x_1(f,t)\rvert}, \frac{1}{2\pi f}\arg\left[\frac{x_2(f,t)}{x_1(f,t)}\right]\right]^{\mathrm{T}}$	[18]	1.9	8.3	14.0
(B)	$\left[\frac{\lvert x_2(f,t)\rvert}{\lvert x_1(f,t)\rvert}, \frac{1}{2\pi f c^{-1} d}\arg\left[\frac{x_2(f,t)}{x_1(f,t)}\right]\right]^{\mathrm{T}}$		5.7	14.1	14.0
(A)′	$\left[\frac{\lvert x_2(f,t)\rvert}{\lvert x_1(f,t)\rvert} - \frac{1}{\frac{\lvert x_2(f,t)\rvert}{\lvert x_1(f,t)\rvert}}, \frac{1}{2\pi f}\arg\left[\frac{x_2(f,t)}{x_1(f,t)}\right]\right]^{\mathrm{T}}$	[3]	1.8	7.9	14.0
(C)	$\frac{1}{2\pi f}\arg\left[\frac{x_2(f,t)}{x_1(f,t)}\right]$		10.5	14.0	14.0
(D)	$\frac{1}{2\pi f c^{-1} d}\arg\left[\frac{x_2(f,t)}{x_1(f,t)}\right]$	[26]	11.6	14.0	14.0
(E)	$\left[\frac{\lvert x_1(f,t)\rvert}{A(f,t)}, \frac{\lvert x_2(f,t)\rvert}{A(f,t)}, \frac{1}{2\pi f}\arg\left[\frac{x_2(f,t)}{x_1(f,t)}\right]\right]^{\mathrm{T}}$		5.2	7.9	14.3
(F)	$\left[\frac{\lvert x_1(f,t)\rvert}{A(f,t)}, \frac{\lvert x_2(f,t)\rvert}{A(f,t)}, \frac{1}{2\pi f c^{-1} d}\arg\left[\frac{x_2(f,t)}{x_1(f,t)}\right]\right]^{\mathrm{T}}$		12.4	14.1	14.2
(G)	$\bar{\Theta}_j(f,t) = \lvert x_j(f,t)\rvert \exp\left[\jmath\,\frac{\arg[x_j(f,t)/x_J(f,t)]}{\alpha_j f}\right]$ $\mathbf{\Theta}(f,t) \leftarrow \bar{\mathbf{\Theta}}(f,t)/\lVert\bar{\mathbf{\Theta}}(f,t)\rVert$		12.2	14.1	14.1

“opt.” shows the performance with the known centroid and two distance measures: the Euclidean distance (ED) (8) and the Mahalanobis distance (MD). $N = 3$, $M = 2$. The performance difference between features C and D was caused by the convergence criteria for the *k*-means. $A(f,t) = \sqrt{\sum_{j=1}^{M} \lvert x_j(f,t)\rvert^2}$. α_j: A weight parameter introduced in Section 4.1.

possible to apply the method to short data without significant performance degradation [27].

3.2. Clustering result with k-means algorithm

Previously, features $\mathbf{\Theta}(f,t)$ were clustered manually [3,18], or with an ML-based gradient method [21]. In contrast, in this subsection, we attempt to employ the well-known *k*-means clustering algorithm [22], which can both automate and simplify the clustering. We show that the previously utilized feature (A) cannot be clustered well by the *k*-means algorithm and provide possible reasons for this.

Table 1 shows the separation performance (the signal to interference ratio (SIR) improvement: see Appendix A) when we cluster each feature with the *k*-means algorithm. In Table 1, we also show the optimal results with known centroid values, which are calculated with known sources and impulse responses (unblind). Here, we utilized two omnidirectional microphones with a 4 cm spacing for three speech sources set at 30°, 70° and 135°, where the distance between the microphones and sources was 50 cm and the room reverberation time was 128 ms. We investigated eight combinations of speakers and averaged the separation results. From Table 1, it can be seen that all features perform similarly when the centroids are known and MD-based clustering is used. Therefore, the separation problem amounts to finding a feature that leads to accurate centroid estimates blindly. However, we can also see that some features (A), (A)′, (B) and (E) cannot achieve good separation performance with the *k*-means.

There are two main reasons for this. The first is related to the outliers of the level ratio $\lvert x_2\rvert/\lvert x_1\rvert$. Fig. 3 shows examples. We can see several large values in the level ratio of (A), although we used omnidirectional microphones where $\lvert x_2\rvert/\lvert x_1\rvert \approx 1$. Due to the outliers in the level ratio, the phase of (A) cannot be clustered (Fig. 3 (A)), although the phase terms themselves can be clustered (Table 1 (C)). This is the reason for the poor performance with (A) and (B). Such outliers occur at too many time-frequency points for them to be removed without degrading the separated sound quality.

We found that when we normalize the level ratios as seen in features (E) and (F), they become $\leqslant 1$ and prevent such outliers (Fig. 3 (F)). Therefore, features (E) and (F) provide better performance than (A) and (B). However, the performance with (E) is still insufficient.

This suggests a second reason: namely that the phase term of feature (A) is too small. This is more important and more fatal than the first reason. For multivariate clustering with the *k*-means algorithm, the level ratios and phase differences should have similar variances. This is because the *k*-means assumes distributions of isotropic variance. However, the phase term of feature (A) is far smaller (Fig. 3 (A)) than the level ratio. The poor

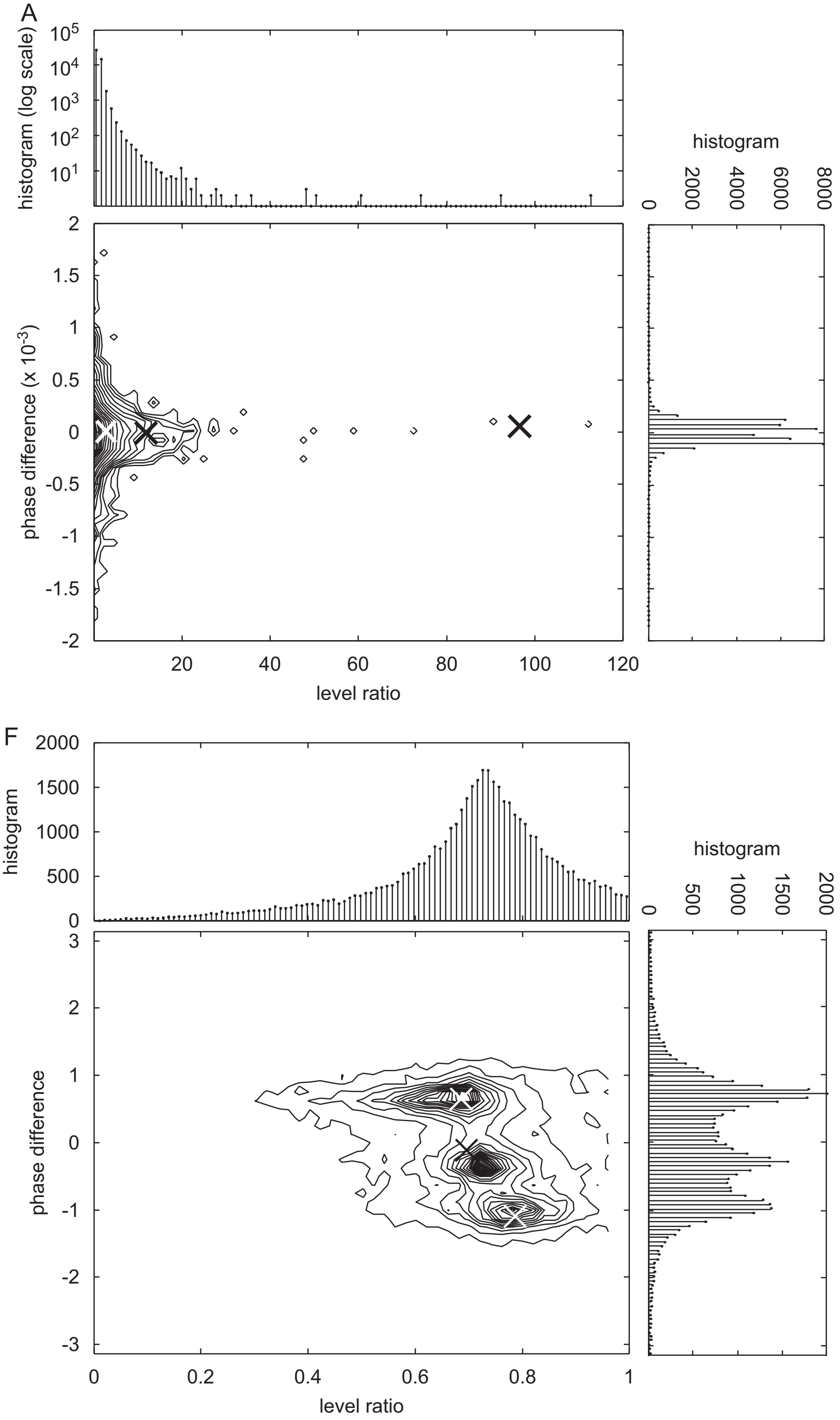

Fig. 3. Example histograms with features (A) and (F). For each feature, top: histogram of the level ratio term of each feature, bottom left: the contour plot of the two-dimensional histogram, bottom right: histogram of the phase difference term of each feature. In the contour plots, × denotes the cluster centroids $\mathbf{c}_k$ obtained by the k-means algorithm. In (F), we plot only two components $[|x_1(f,t)|/A(f,t),\ 1/2\pi fc^{-1}d\ \arg[x_2(f,t)/x_1(f,t)]]^{\mathrm{T}}$.

Table 2
Performance (SIR improvement in dB and SDR in dB: see Appendix A for the definitions) and computation time with the k-means algorithm and the GMM fitting

	k-means			GMM		
Feature	SIR imprv.	SDR	time (s)	SIR imprv.	SDR	time (s)
(A)	1.9	7.5	5.0	11.8	9.4	26.3
(A)$'$	1.8	7.5	7.1	11.3	9.8	26.3
(E)	5.2	6.0	4.6	12.8	8.8	33.7
(F)	12.4	10.3	2.5	14.2	9.7	26.5

$N = 3$, $M = 2$. GMM fitting needed 6 Gaussians.

performance with (A) and (E) result from this imbalance between the level ratio and phase difference terms. With features (B) and (F), where the phase is divided by $2\pi fc^{-1}d$, the phase difference becomes larger (see Fig. 3 (F)). Therefore, feature (F), where both the level ratios and phase difference are normalized appropriately, achieves good performance (see Table 1) with the k-means algorithm.

It should be noted that if we can handle the different variance with, for example the Gaussian mixture model (GMM) fitting, features (A) and (E) can also work as shown in Table 2. In our experiments, the GMM fitting with only $N = 3$ Gaussians did not work. We needed 6 Gaussians for the successful fitting, and therefore, utilized the posteriors of 3 dominant Gaussians as separation masks. This selection of the number of Gaussians required a lot of trial and error. Furthermore, it should be noted that we had to set appropriate initial values for the mean and variance of the Gaussians carefully for the GMM fitting. If the selection is successful, the GMM fitting works. If not, it does not work at all. Table 2 shows that the appropriate GMM can achieve reasonable performance even for features (A) and (E). From Table 2, we can say that the reason for the poor performance with the k-means for features (A) and (E) arises from the different variances of the level ratio and the phase difference.

We would like to note that the GMM fitting needs sufficient computational time. Table 2 also gives the calculation time with the k-means and GMM fitting. Here, we separated mixtures of 5 s with AthlonXP 3200+ and MATLAB 6.5. The calculation time was measured with the `cputime` command of MATLAB. The computation time provided in Table 2 is the averaged result for eight speaker combinations. We can see from Table 2 that the k-means with feature (F) achieves sufficiently high performance with a shorter computational time than realized with the GMM fitting. The small computational cost of the k-means is attractive.

In conclusion, we found that feature (F) provides more accurate clustering result than other features when either the GMM fitting or the k-means clustering are employed. Moreover, clustering is successfully and effectively executed with the k-means, when we use normalized level ratios and phase differences such as feature (F) (see Fig. 3 (F) and Table 1 (F)).

4. Proposed new feature

Based on the clustering results described in the previous section, we propose a new feature that can be clustered by the k-means algorithm. We also extend the feature to a multiple sensor version, where we can utilize more than three sensors arranged non-linearly to separate two- or three-dimensionally located sources. As our method can be considered as an extension of the DUET, we call our method Multiple sENsor dUET: MENUET.

Our proposed feature has the same property as feature (F) in Section 3, that is, the level ratios and phase differences are appropriately normalized. Our new feature also has a parameter that controls the weight for the level ratios and phase differences. Moreover, our normalization does not require sensor position information. This allows us to apply our method to an arbitrarily arranged sensor array.

Because the basic scheme is the same as that in Fig. 1, here we focus mainly on our new feature vector.

4.1. New feature

Our new feature employs the normalized level ratios and phase differences between multiple observations:

$$\boldsymbol{\Theta}(f,t) = [\boldsymbol{\Theta}^L(f,t), \boldsymbol{\Theta}^P(f,t)]^{\mathrm{T}}, \tag{15}$$

where

$$\boldsymbol{\Theta}^L(f,t) = \left[\frac{|x_1(f,t)|}{A(f,t)}, \ldots, \frac{|x_M(f,t)|}{A(f,t)}\right], \tag{16}$$

$$\boldsymbol{\Theta}^P(f,t) = \left[\frac{1}{\alpha_1 f}\arg\left[\frac{x_1(f,t)}{x_J(f,t)}\right], \ldots, \frac{1}{\alpha_M f}\arg\left[\frac{x_M(f,t)}{x_J(f,t)}\right]\right] \tag{17}$$

In the above equations, $A(f,t) = \sqrt{\sum_{j=1}^{M} |x_j(f,t)|^2}$, J is the index of one of the sensors, and α_j ($j = 1, \ldots, M$) is a positive weighting constant. By changing α_j, we can control the weights for the level ratio and phase difference information of the observed signals; a larger value puts weight on the level ratio and a smaller value emphasizes the phase difference.

The normalized level ratio has the property of $0 \leqslant \Theta_j^L(f,t) \leqslant 1$, where Θ_j^L is the jth component of $\mathbf{\Theta}_L$. This can prevent the outliers discussed in the previous section.

An appropriate value for the phase weight is $\alpha_j = \alpha = 4\pi c^{-1} d_{\max}$, where c is the propagation velocity and $d_{\max}$ is the maximum distance[1] between sensor J and sensor $^\forall j \in \{1, \ldots, M\}$. Let us provide the reason for this. Here, we use the mixing model (13) and, without loss of generality, we assume that the delay parameter τ_{jk} is determined by the path difference $l_{jk} - l_{Jk}$:

$$\tau_{jk} = (l_{jk} - l_{Jk})/c, \tag{18}$$

where l_{jk} is the distance from source k to sensor j. We also use the fact that the maximum distance $d_{\max}$ between the sensors is greater than the maximum path difference:

$$\max_{j,k} |l_{jk} - l_{Jk}| \leqslant d_{\max}.$$

Using these assumptions and the mixing model (13), $\Theta_j^P(f,t)$, which is the jth component of $\mathbf{\Theta}_P(f,t)$, becomes

$$\frac{1}{\alpha_j f} \arg\left[\frac{x_j(f,t)}{x_J(f,t)}\right] = \frac{2\pi c^{-1}(l_{jk} - l_{Jk})}{\alpha_j} \leqslant \frac{2\pi c^{-1} d_{\max}}{\alpha_j}. \tag{19}$$

Because the level ratio is normalized to have the range $0 \leqslant \mathbf{\Theta}^L(f,t) \leqslant 1$, the phase difference $\Theta_j^P(f,t)$ should also be normalized so that it has a similar range. If we allow $\Theta_j^P(f,t)$ to have the range $-\frac{1}{2} \leqslant \mathbf{\Theta}^P(f,t) \leqslant \frac{1}{2}$ (note that $\Theta_j^P(f,t)$ can take a negative value), we have equality in (19) when $\alpha_j = \alpha = 4\pi c^{-1} d_{\max}$. That is, $\alpha = 4\pi c^{-1} d_{\max}$ realizes the same range width as that of the level ratio.

4.2. Modified proposed feature

We can modify our proposed new feature (15) by using the complex representation,

$$\Theta_j(f,t) = \Theta_j^L(f,t) \exp[\jmath \Theta_j^P(f,t)], \tag{20}$$

where Θ_j^L and Θ_j^P are the jth components of (16) and (17). This modification can also be realized by [28]

$$\bar{\Theta}_j(f,t) = |x_j(f,t)| \exp\left[\jmath \frac{\arg[x_j(f,t)/x_J(f,t)]}{\alpha_j f}\right], \tag{21}$$

$$\mathbf{\Theta}(f,t) \leftarrow \bar{\mathbf{\Theta}}(f,t)/\|\bar{\mathbf{\Theta}}(f,t)\|, \tag{22}$$

where $\bar{\mathbf{\Theta}}(f,t) = [\bar{\Theta}_1(f,t), \ldots, \bar{\Theta}_M(f,t)]^{\mathrm{T}}$. Feature (22) is our modified feature, where the phase difference information is held in the argument term (21), and the level ratio is normalized by the vector norm normalization (22). The weight parameter α_j has the same property as (15); however, $\alpha = 4c^{-1} d_{\max}$ should be the lower limit for successful clustering (see Appendix B).

Now the normalized vectors $\mathbf{\Theta}(f,t)$ (22) are M-dimensional complex vectors, and therefore the clustering of the features will be carried out in an M-dimensional complex space. The unit-norm normalization (22) makes the distance calculation in the clustering (7) easier, because it projects the vector on a hyper unit sphere. If the features $\mathbf{\Theta}(f,t)$ and the cluster centroid $\mathbf{c}_k$ are on the unit sphere, i.e., $\|\mathbf{\Theta}(f,t)\| = \|\mathbf{c}_k\| = 1$, the square distance $\|\mathbf{\Theta}(f,t) - \mathbf{c}_k\|^2 = 2(1 - \mathrm{Re}(\mathbf{c}_k^H \mathbf{\Theta}(f,t)))$. That is, the minimization of the distance $\|\mathbf{\Theta}(f,t) - \mathbf{c}_k\|^2$ is equivalent to the maximization of the real part of the inner product $\mathbf{c}_k^H \mathbf{\Theta}(f,t)$, whose calculation is less demanding in terms of computational complexity.

5. Experiments

5.1. Experimental conditions

We performed experiments with measured impulse responses $\mathrm{h}_{jk}(l)$ in a room as shown in Figs. 4 and 5. The room reverberation times RT_{60} were 128 and 300 ms. We used the same room for both reverberation times but changed the wall condition. We also changed the distance R between the sensors and sources. The distance variations were $R = 50$, 110, and 170 cm (see Figs. 4 and 5). Mixtures were made by convolving the measured impulse responses in the room and 5-s English speeches. For the anechoic test, we simulated the mixture by using the anechoic model ((13) and (18)) and the mixture model (1). The sampling rate was 8 kHz. The STFT frame size T was 512 and the window shift was $T/4$.

Unless otherwise noted, we utilized modified feature (22) with $\alpha_j = \alpha = 4c^{-1} d_{\max}$ for the features, because the computational cost of distance

[1] If we do not have an accurate value for $d_{\max}$, we may use a rough positive constant, as shown in Section 5.2.4.

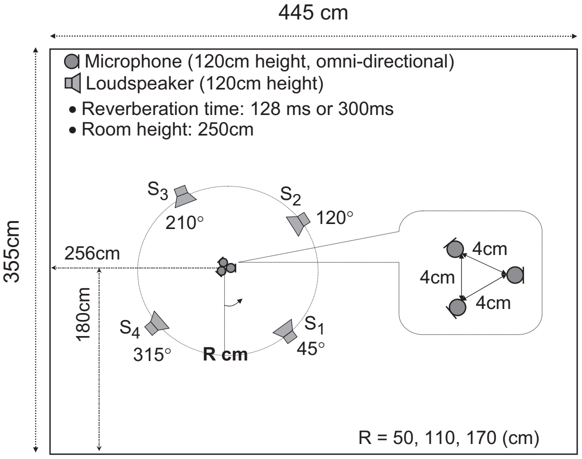

Fig. 4. Room setup ($N = 4, M = 3$).

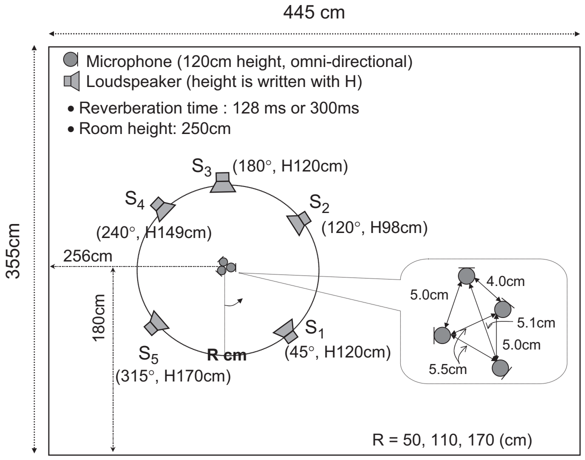

Fig. 5. Room setup ($N = 5, M = 4$).

calculation is lowered (see Section 4.2). We utilized the k-means algorithm for the clustering, where the number of sources N was given. We set the initial centroids of the k-means using the far-field model where the frequency response $h_{jk}(f)$ is given as $h_{jk}(f) \approx \exp[-\jmath 2\pi f c^{-1} \mathbf{d}_j^{\mathrm{T}} \mathbf{q}_k]$, and using the same normalization as each feature. Here, c is the propagation velocity of the signals, and the three-dimensional vectors $\mathbf{d}_j$ and $\mathbf{q}_k$ represent the location of sensor j and the direction of source k, respectively [29]. The sensor locations $\mathbf{d}_j$ ($j = 1, \ldots, M$) were on almost the same scale in each setup, and the initial directions $\mathbf{q}_k$ were set so that they were as scattered as possible. Concretely, we utilized the sensor vector $\mathbf{q}_k = [\cos\theta_k \cos\phi_k, \sin\theta_k \cos\phi_k, \sin\phi_k]^{\mathrm{T}}$ where the azimuth of kth source was set as $\theta_k = \Pi \times k$ ($k = 1, \ldots, N$, $\Pi = 2\pi/N$ for $M \geqslant 3$ and $\Pi = \pi/N$ for $M = 2$), and the elevation $\phi_k = 0$ for all sources k. Note that these initial values of $\boldsymbol{d}_j$ and $\mathbf{q}_k$ were not exactly the same as those in each setup.

The separation performance was evaluated in terms of the SIR improvement and the signal to distortion ratio (SDR). Their definitions are found in Appendix A. We investigated eight combinations of speakers and averaged the separation results.

5.2. *Separation results*

5.2.1. *With two sensors*

First, we tested our feature with two sensors under the condition described in Section 3, and compared the result with that of previous features. Table 1 in Section 3.2 shows the result. The proposed feature (15) corresponds to (F) and the modified feature (22) is shown as (G). We obtained better separation performance with our proposed features than with other features (A)–(E). A comparison of the performance achieved with our proposed method and with the GMM fitting is shown in Table 2. The comparison was investigated only for the two sensor case. Our proposed feature (F) achieves high performance with the k-means within a shorter computation time than with the GMM fitting. Moreover, we can see that our proposed feature (F) is also suitable for the GMM fitting. A comparison with the MAP approach can be found in [16]. In [16], it is shown that the proposed method yields better performance in terms of SIR than the MAP approach. It is also pointed out that proposed method causes larger non-linear distortion in its outputs than the MAP approach.

We also compared our proposed method with the DESPRIT algorithm [24], using a linear array of three microphones for four sources. It should be noted that the previous DESPRIT limits its array shape to a linear array or two sets of congruent arrays, as discussed in Section 1. In the experiments, we did not see big difference in performance between our MENUET and the DESPRIT. That is, the proposed MENUET also works with a linear array (i.e., one-dimensional array).

Note that two sensors/linear arrays do not work when the sources are placed at axisymmetrical locations with respect to the microphone array, because they have the equivalent features in (12).

5.2.2. *With two-dimensional three sensors*

Here, we show the separation results obtained with three sensors arranged two-dimensionally

(Fig. 4). Note that sources were also distributed two-dimensionally.

Fig. 6(a) shows the separation result when $N = 4$ and $M = 3$. We can see that our proposed method achieved good separation performance with the non-linear sensor arrangement. We also evaluated the performance for $N = 5, M = 3$, where the source positions were 45°, 120°, 210°, 280° and 345°, and obtained good performance (Fig. 6(b)).

5.2.3. *With four sensors*

We also applied our method to a three-dimensional sensor array arranged non-uniformly (Fig. 5). Here, the system knew only the maximum distance $d_{\max}$ (5.5 cm) between the reference microphone and the others. To avoid the spatial aliasing problem, we utilized frequency bins up to 3100 Hz in this setup. Fig. 6(c) shows the separation result when $N = 5$ and $M = 4$. Fig. 6(c) shows that our proposed new feature can be applied to such three-dimensional microphone array systems.

5.2.4. *Weight parameter α*

Here, we examine the relationship between the phase weight parameter α and the separation performance. As mentioned in Section 4.1, when α is large the level ratio is emphasized, and when α is small the phase difference is emphasized. Fig. 7 shows the relationship when $N = 4$ and $M = 3$ (Fig. 4) with the proposed feature (15) and the modified feature (22). Note that $\alpha = 2\pi$ corresponds to the previous feature (A) (Table 1).

Fig. 7(a) shows the result with the proposed feature (15). It achieved high performance when α was sufficiently small. This is because the phase difference between the sensors was more reliable than the level ratio, due to our microphone setup. As α became small, the performance saturated. On the other hand, the performance degraded as α became large. This is caused by the imbalance between level ratio and phase difference terms, because the phase term becomes too small when α becomes large.

With modified feature (22), we obtained the best performance around $\alpha = 4c^{-1}d_{\max}$ (Fig. 7(b)). This is because $\alpha = 4c^{-1}d_{\max}$ realizes the full use of the phase difference information (Appendix B), which is preferable for our sensor setting. As α became large, the performance degraded. When $\alpha < 4c^{-1}d_{\max}$ the performance also worsened. It should be remembered that, with the modified feature, $\alpha = 4c^{-1}d_{\max}$ should be the lower limit (see Section 4.2). When $\alpha < 4c^{-1}d_{\max}$, the distance measure (7) for the clustering is not evaluated correctly (see Appendix B), and therefore, the clustering stage failed and the performance worsened.

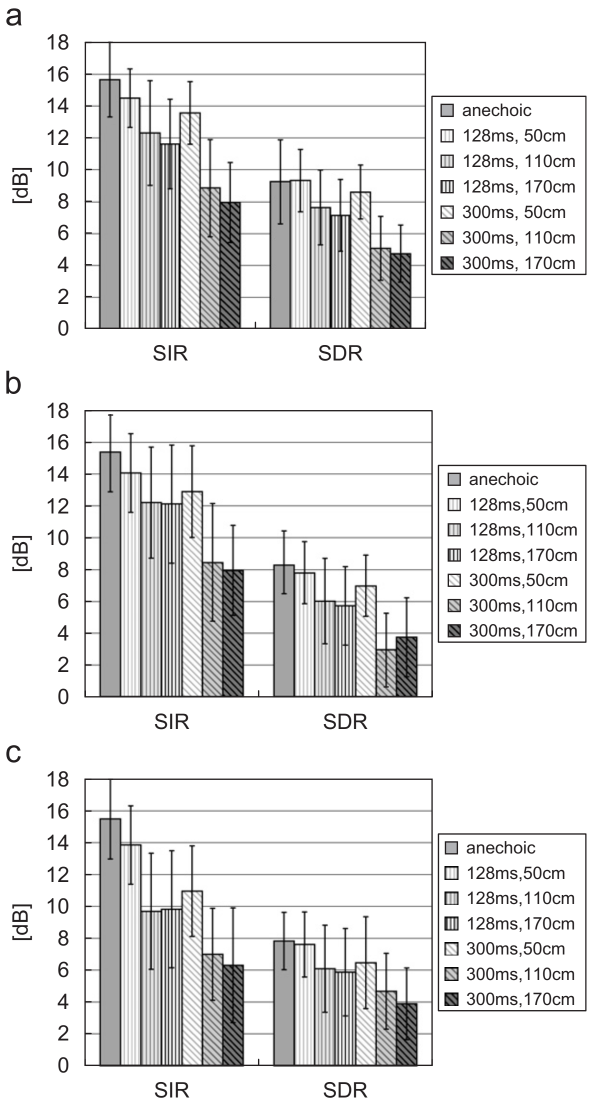

Fig. 6. Average SIR improvement and SDR for each condition. The error bar shows the standard deviation for all outputs and combinations: (a) $N = 4, M = 3$ (average input SIR ≈ -4.8 [dB]); (b) $N = 5, M = 3$ (average input SIR ≈ -6.0 [dB]); (c) $N = 5, M = 4$ (average input SIR ≈ -6.3 [dB]).

We can also see from Fig. 7 that both proposed features (15) and (22) achieved good performance over a wide α range. This means that we do not need the exact maximum sensor spacing $d_{\max}$. This allows us to utilize an arbitrarily arranged sensor array, although similar distances between pairs of sensors

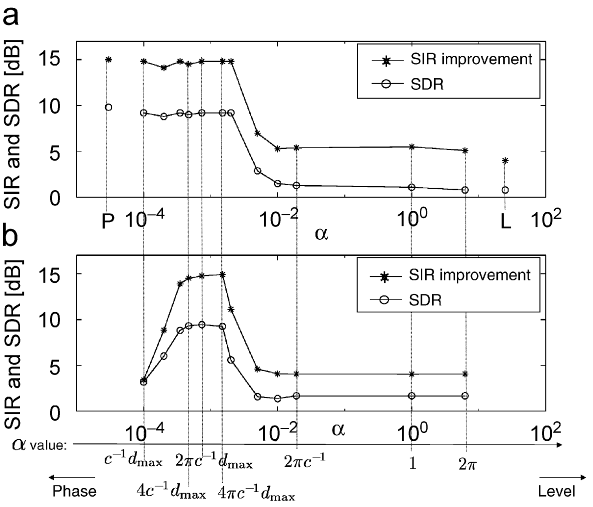

Fig. 7. Relationship between α and separation performance when $N = 4$, $M = 3$, $R = 50\,\mathrm{cm}$, and $\mathrm{RT}_{60} = 128\,\mathrm{ms}$. (a) with feature (15) and (b) with modified feature (22). In (a), "P" denotes the performance with $\boldsymbol{\Theta} = \boldsymbol{\Theta}^P$, and "L" means with $\boldsymbol{\Theta} = \boldsymbol{\Theta}^L$.

are preferable so that the k-means can use all sensor information optimally.

5.3. Discussion

Fig. 6 also shows the performance tendency in reverberant conditions. The performance degrades as the reverberation time RT_{60} becomes long. Moreover, performance degradation was observed as the distance R became large. This is because, under long reverberation and/or large distance R conditions, the direct sound contribution to the impulse responses becomes smaller, and the source sparseness (5) and anechoic assumptions (13) cannot hold.

We assessed the way in which reverberation time and distance R affect the source sparseness (5) and anechoic assumptions (13). Table 3 shows the average clarity index C ([30] and Appendix C), which explains the ratio between direct sound and reverberant sound. Small (large) C means the reverberant sound (direct sound) is large. We can see from Table 3 that the clarity index C becomes small as the reverberation time and distance R increase. That is, when the reverberation time is long and distance R is large, the anechoic assumption (13) seems to become corrupted. For the sparseness measure, we employed the approximate W-disjoint orthogonality r_k ([3] and Appendix C). Fig. 8 shows the approximate W-disjoint orthogonality under some reverberant conditions. As the sparseness increases the approximated W-disjoint orthogonality r_k increases, and vice versa. As seen in Fig. 8, the sparseness decreases with increases in both the reverberation time and distance R. That is, the sparseness decreases when the contribution of the direct sound is small (see Table 3). In addition, we can see that an increase in the number of sources also reduces the sparseness.

Table 3
Average clarity index [dB]

	$R = 50\,\mathrm{cm}$	$R = 110\,\mathrm{cm}$	$R = 170\,\mathrm{cm}$
$\mathrm{RT}_{60} = 128\,\mathrm{ms}$	45.5	40.8	36.6
$\mathrm{RT}_{60} = 300\,\mathrm{ms}$	40.5	34.9	32.7

It is also important to mention non-linear distortion in separated signals. There is non-linear distortion (musical noise) in the outputs with our method, just as there is in the outputs with the previous binary mask approaches. The results of subjective tests with 10 listeners can be found in [28]. Some sound examples can be found at [31].

6. Conclusion

We proposed a novel sparse source separation method (MENUET) based on the normalization and clustering of the level ratios and phase differences between multiple observations. Our proposed features can effectively employ the level ratios and phase differences, and are clustered easily by the well-known k-means algorithm. It should be noted that the k-means is optimal when the clusters are *Gaussian*; however, this is not always true even for our proposed feature (F) (see Fig. 3 (F)). However, as shown in this paper, the proposed feature with the k-means achieved sufficiently high performance. Moreover, our feature makes it easy to employ multiple sensors arranged in a non-linear/non-uniform way. We obtained promising experimental results in a room with weak reverberation even under underdetermined conditions. Although we provided results solely for underdetermined cases in this paper, our proposed method can also be applied to (over-) determined cases [28].

We also reported the separation performance under some reverberant conditions, where the sparseness and anechoic assumptions were deteriorating. From the results, we saw that the direct and reverberant ratio is important for the current sparse

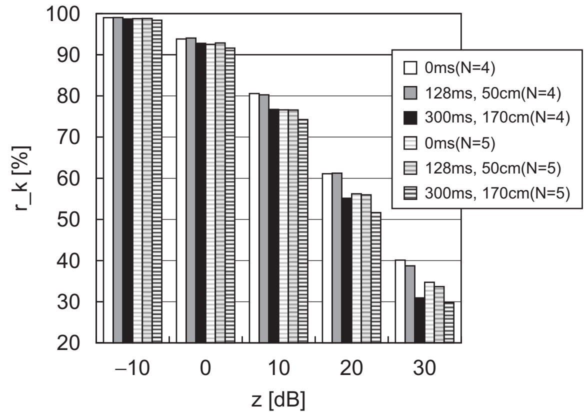

Fig. 8. Source sparseness $r_k(z)$ for some reverberant conditions.

source separation. The sparse source separation in reverberant conditions is still an open problem.

Appendix A. Performance measures

The SIR improvement was calculated by

$\text{OutputSIR}_i - \text{InputSIR}_i$,

where

$$\text{InputSIR}_i = 10\log_{10}\frac{\sum_t|\mathrm{x}_{Ji}(t)|^2}{\sum_t|\sum_{k\neq i}\mathrm{x}_{Jk}(t)|^2}(\mathrm{dB}), \tag{A.1}$$

$$\text{OutputSIR}_i = 10\log_{10}\frac{\sum_t|\mathrm{y}_{ii}(t)|^2}{\sum_t|\sum_{k\neq i}\mathrm{y}_{ik}(t)|^2}(\mathrm{dB}), \tag{A.2}$$

where $\mathrm{x}_{Jk}(t) = \sum_l \mathrm{h}_{Jk}(l)\mathrm{s}_k(t-l)$ and $\mathrm{y}_{ik}(t)$ is the component of s_k that appears at output $\mathrm{y}_i(t)$: $\mathrm{y}_i(t) = \sum_{k=1}^{N}\mathrm{y}_{ik}(t)$.

The SDR is employed to evaluate the sound quality:

$$\text{SDR}_i = 10\log_{10}\frac{\sum_t|\mathrm{x}_{Ji}(t)|^2}{\sum_t|\mathrm{x}_{Ji}(t)-\beta\mathrm{y}_{ii}(t-D)|^2}(\mathrm{dB}), \tag{A.3}$$

where β and D are parameters used to compensate for the amplitude and phase difference between x_{Ji} and y_{ii}.

Appendix B. Value of α for modified feature (21)

In this section, we show the required condition for the phase weight parameter α for modified feature (21). Because the modified feature (22) is a complex vector, we have to consider the phase term when we perform clustering. When α in (21) is too large, the variance of the phase term becomes smaller than that of the level term. On the other hand, when α in (21) is too small, the phase changes too fast and causes a kind of aliasing problem. Moreover, it is important that the distance measure (7) of the clustering holds the condition: $|\Theta - \Theta'|$ increases monotonically as $|\arg[\Theta] - \arg[\Theta']|$ increases. However, if the phase term is larger than $\pi/2$, such a monotonic increase cannot hold. That is the phase term should have the following relationship:

$$-\frac{\pi}{2} \leqslant \arg[\Theta] \leqslant \frac{\pi}{2}. \tag{B.1}$$

Let us model the mixing process as (13) and, without loss of generality, we assume that the delay parameter τ_{jk} is determined by the path difference $l_{jk} - l_{Jk}$:

$$\tau_{jk} = (l_{jk} - l_{Jk})/c, \tag{B.2}$$

where l_{jk} is the distance from source k to sensor j. This assumption makes $\tau_{Jk} = 0$. Substituting the mixing model (B.2) and (13), and the sparseness assumption (5) into (21) and (22) yields

$$\mathbf{\Theta}(f,t) \approx \frac{\lambda_{jk}}{D_k}\exp\left[-\jmath\frac{2\pi c^{-1}(l_{jk}-l_{Jk})}{\alpha_j}\right], \tag{B.3}$$

where $D_k = \sqrt{\sum_{j=1}^{M}\lambda_{jk}^2}$.

From the condition (B.1) and Eq. (B.3), the lower limit of α is given as

$$\begin{aligned}|\arg[\Theta]| &= \left|\frac{2\pi c^{-1}(l_{jk}-l_{Jk})}{\alpha_j}\right| \\ &\leqslant \left|\frac{2\pi c^{-1}d_{\max}}{\alpha_j}\right| \leqslant \frac{\pi}{2}, \qquad \text{(B.4)} \\ \alpha_j &\geqslant 4c^{-1}d_{\max}. \qquad \text{(B.5)}\end{aligned}$$

In (B.4), we used the fact that $\max_{j,k}|l_{jk} - l_{Jk}| \leqslant d_{\max}$.

From (B.5), we can conclude that the phase parameter $\alpha = 4c^{-1}d_{\max}$ should be the minimum value to maintain the relationship (B.1). In addition (B.1) has equality when $\alpha = 4c^{-1}d_{\max}$, which means that the phase difference information is most scattered. That is, the weight with $\alpha = 4c^{-1}d_{\max}$ allows us to make full use of the phase difference information. This is a preferable property for small sensor array systems (e.g., see Section 5), where phase differences between sensors are more reliable than level ratios for clustering.

Appendix C. Measures for reverberation and sparseness assessments

The clarity index [30]

$$C = 10\log_{10}\frac{\int_0^{80\,\mathrm{ms}} h^2(t)\,\mathrm{d}t}{\int_{80\,\mathrm{ms}}^{\infty} h^2(t)\,\mathrm{d}t}(\mathrm{dB})$$

explains the ratio between direct and reverberant sound. Small (large) C means the reverberant sound (direct sound) is large.

The sparseness measure, that is the approximate W-disjoint orthogonality, is defined as [3]

$$r_k(z) = \frac{\sum_{(f,t)}\|\Phi_{(k,z)}(f,t)s_k(f,t)\|^2}{\sum_{(f,t)}\|s_k(f,t)\|^2} \times 100(\%), \qquad \text{(C.1)}$$

where $\Phi_{(k,z)}$ is a time-frequency binary mask that has a parameter z

$$\Phi_{(k,z)}(f,t) = \begin{cases} 1 & 20\log(|s_k(f,t)|/|\hat{y}_k(f,t)|) > z, \\ 0 & \text{otherwise} \end{cases} \qquad \text{(C.2)}$$

and $\hat{y}_k(f,t) = \mathrm{STFT}[\sum_{i=1,i\neq k}^{N} s_i(t)]$ (sum of interference components). The approximate W-disjoint orthogonality $r_k(z)$ indicates the percentage of the energy of source k for time–frequency points where it dominates the other sources by z dB. A larger approximate W-disjoint orthogonality $r_k(z)$ means more sparseness, and vice versa.

References

[1] S. Haykin (Ed.), Unsupervised Adaptive Filtering (Volume I: Blind Source Separation), Wiley, New York, 2000.

[2] A. Hyvärinen, J. Karhunen, E. Oja, Independent Component Analysis, Wiley, New York, 2001.

[3] Ö. Yılmaz, S. Rickard, Blind separation of speech mixtures via time–frequency masking, IEEE Transactions on SP 52 (7) (2004) 1830–1847.

[4] H. Buchner, R. Aichner, W. Kellermann, Blind source separation for convolutive mixtures: a unified treatment, in: Y. Huang, J. Benesty (Eds.), Audio Signal Processing for Next-Generation Multimedia Communication Systems, Kluwer Academic Publishers, Dardrecht, 2004, pp. 255–293.

[5] H. Sawada, R. Mukai, S. Araki, S. Makino, Frequency-domain blind source separation, in: J. Benesty, S. Makino, J. Chen (Eds.), Speech Enhancement, Springer, Berlin, 2005, pp. 299–327.

[6] S. Amari, S. Douglas, A. Cichocki, H. Yang, Multichannel blind deconvolution and equalization using the natural gradient, in: Proceedings of IEEE Workshop on Signal Processing Advances in Wireless Communications, 1997, pp. 101–104.

[7] P. Smaragdis, Blind separation of convolved mixtures in the frequency domain, Neurocomputing 22 (1998) 21–34.

[8] L. Parra, C. Spence, Convolutive blind separation of non-stationary sources, IEEE Trans. Speech Audio Process. 8 (3) (2000) 320–327.

[9] J. Anemüller, B. Kollmeier, Amplitude modulation decorrelation for convolutive blind source separation, in: Proceedings of the ICA 2000, 2000, pp. 215–220.

[10] S. Araki, R. Mukai, S. Makino, T. Nishikawa, H. Saruwatari, The fundamental limitation of frequency domain blind source separation for convolutive mixtures of speech, IEEE Trans. Speech Audio Process. 11 (2) (2003) 109–116.

[11] F. Theis, E. Lang, C. Puntonet, A geometric algorithm for overcomplete linear ICA, Neurocomputing 56 (2004) 381–398.

[12] P. Bofill, M. Zibulevsky, Blind separation of more sources than mixtures using sparsity of their short-time Fourier transform, in: Proceedings of the ICA2000, 2000, pp. 87–92.

[13] L. Vielva, D. Erdogmus, C. Pantaleon, I. Santamaria, J. Pereda, J.C. Principe, Underdetermined blind source separation in a time-varying environment, in: Proceedings of the ICASSP 2002, 2002, pp. 3049–3052.

[14] P. Bofill, Underdetermined blind separation of delayed sound sources in the frequency domain, Neurocomputing 55 (2003) 627–641.

[15] A. Blin, S. Araki, S. Makino, Underdetermined blind separation of convolutive mixtures of speech using time–frequency mask and mixing matrix estimation, IEICE Trans. Fundam. E88-A (7) (2005) 1693–1700.

[16] S. Winter, W. Kellermann, H. Sawada, S. Makino, MAP-based underdetermined blind source separation of convolutive mixtures by hierarchical clustering and $l1$-norm minimization, EURASIP J. Adv. Signal Process. (2007), Article ID 24717.

[17] J.M. Peterson, S. Kadambe, A probabilistic approach for blind source separation of underdetermined convolutive mixtures, in: Proceedings of the ICASSP 2003, vol. VI, 2003, pp. 581–584.

[18] A. Jourjine, S. Rickard, Ö. Yılmaz, Blind separation of disjoint orthogonal signals: demixing N sources from 2 mixtures, in: Proceedings of the ICASSP 2000, vol. 12, 2000, pp. 2985–2988.

[19] M. Aoki, M. Okamoto, S. Aoki, H. Matsui, T. Sakurai, Y. Kaneda, Sound source segregation based on estimating incident angle of each frequency component of input signals acquired by multiple microphones, Acoust. Sci. Technol. 22 (2) (2001) 149–157.

[20] N. Roman, D. Wang, G.J. Brown, Speech segregation based on sound localization, J. Acousit. Soc. Am. 114 (4) (2003) 2236–2252.

[21] S. Rickard, R. Balan, J. Rosca, Real-time time–frequency based blind source separation, in: Proceedings of the ICA 2001, 2001, pp. 651–656.

[22] R.O. Duda, P.E. Hart, D.G. Stork, Pattern Classification, second ed., Wiley Interscience, New York, 2000.

[23] R. Balan, J. Rosca, S. Rickard, Non-square blind source separation under coherent noise by beamforming and time–frequency masking, in: Proceedings of the ICA 2003, 2003, pp. 313–318.

[24] T. Melia, S. Rickard, C. Fearon, Histogram-based blind source separation of more sources than sensors using a DUET-ESPRIT technique, in: Proceedings of the EUSIPCO 2005, 2005.

[25] S. Araki, S. Makino, H. Sawada, R. Mukai, Reducing musical noise by a fine-shift overlap-add method applied to source separation using a time–frequency mask, in: Proceedings of the ICASSP 2005, vol. III, 2005, pp. 81–84.

[26] S. Araki, S. Makino, A. Blin, R. Mukai, H. Sawada, Underdetermined blind separation for speech in real environments with sparseness and ICA, in: Proceedings of the ICASSP 2004, vol. III, 2004, pp. 881–884.

[27] S. Araki, H. Sawada, R. Mukai, S. Makino, Normalized observation vector clustering approach for sparse source separation, in: Proceedings of the EUSIPCO 2006, 2006.

[28] S. Araki, H. Sawada, R. Mukai, S. Makino, A novel blind source separation method with observation vector clustering, in: Proceedings of the 2005 International Workshop on Acoustic Echo and Noise Control (IWAENC 2005), 2005, pp. 117–120.

[29] S. Araki, H. Sawada, R. Mukai, S. Makino, DOA estimation for multiple sparse sources with normalized observation vector clustering, in: Proceedings of the ICASSP 2006, vol. 5, 2006, pp. 33–36.

[30] ISO 3382: Acoustics-measurement of the reverberation time of rooms with reference to other acoustical parameters (1997).

[31] ⟨http://www.kecl.ntt.co.jp/icl/signal/araki/xcluster_fine.html⟩.