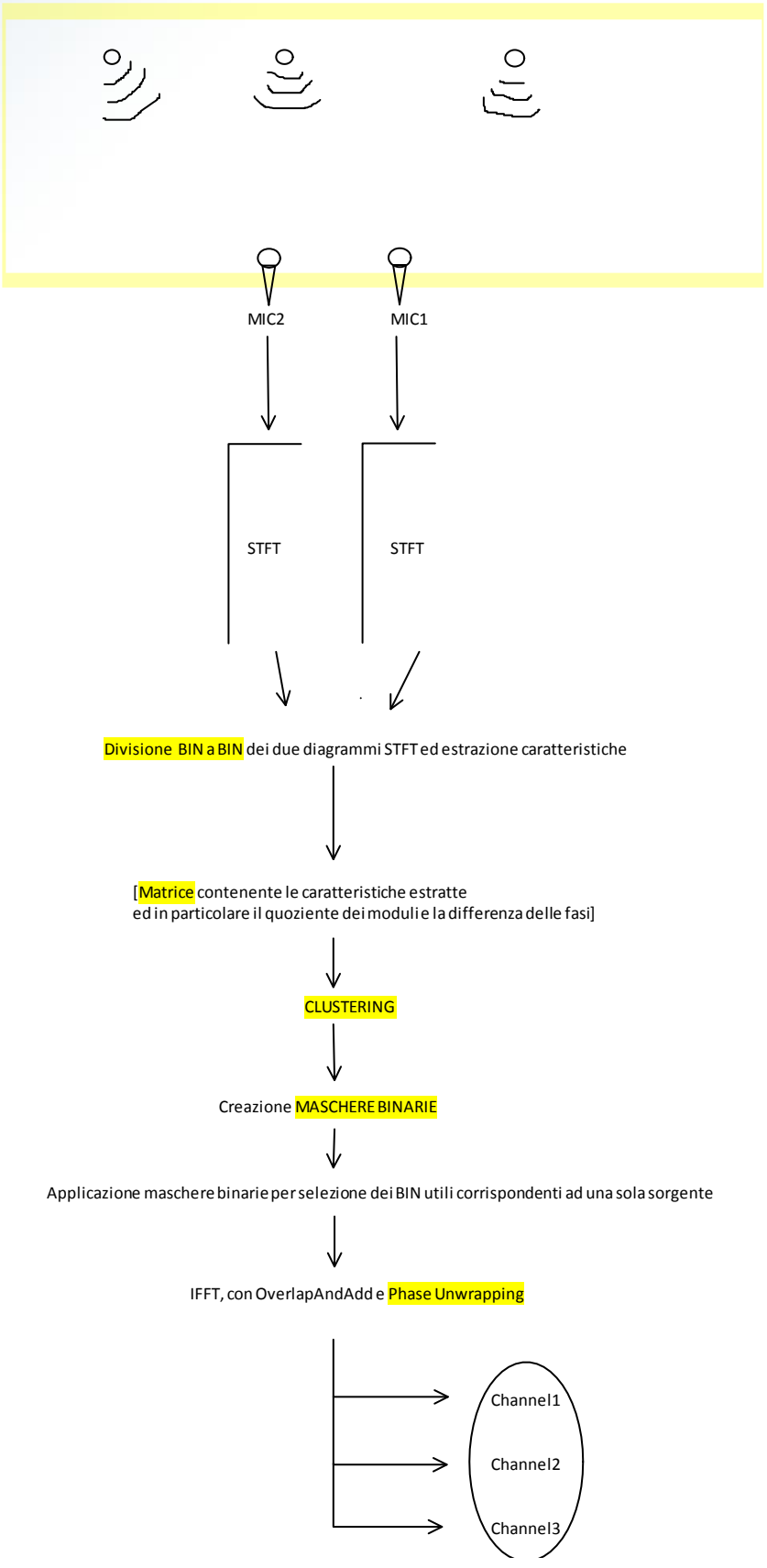
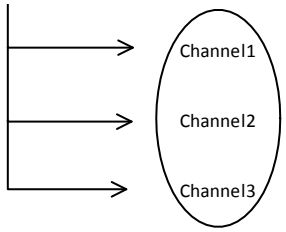
Tutte le pagine seguenti si riferiscono a questo schema

Schema a blocchi:





File utilizzati per simulare le sorgenti --> Potrebbero essere utilizzate tre vocali per la simulazione di tre sorgenti --> in questo caso si avrebbe la certezza della SPARSITà; sarebbe una prima prova per valutare il corretto funzionamento delle procedure

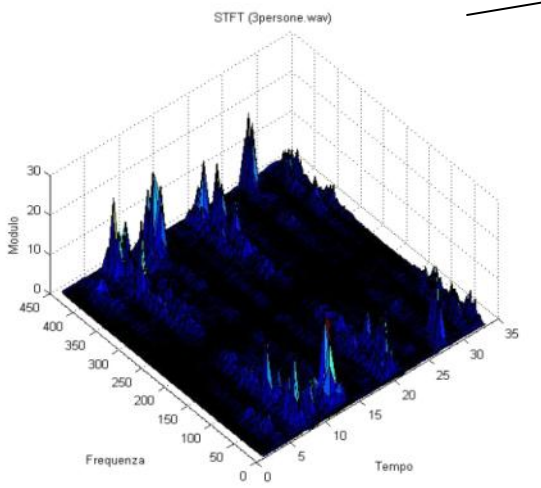
Algoritmi

lunedì 23 marzo 2009
13.46

Codice Antonacci ULA.m

Chiaramente per la simulzione devono essere creati anche due file diversamente ritardati per ogni sorgente per simulare le ricezioni diverse dei due distinti microfoni

STFT



Come descritto nell'articolo 1

-->STFT , size della finestra T=512 con fs=8kHz ,overlap di T/4 (nell'articolo ...capire se va bene T/2) -->T/2 !!!
-->sorgenti sonore di durata circa 3 sec
-->valutazioni fatte con SIR e SDR (vedere l'appendice del primo articolo!) ???

```
%FILE DEL PROGETTO DI TATA
%prova stft e risintesi di un segnale audio
%
% CALCOLO DEL NUMERO DI SPLICE:
% numero_splice=1+((length(signal) -length(win))/(length(win)/2))
%


clear all;
close all;

%creazione finestra
win=hamming(512);

%definita anche la frequenza di campionamnto per come viene
importato il file audio

[signal1,Fs1]=wavexread('3personePCM.wav');
[signal2,Fs2]=wavexread('Toms_diner.wav');
[signal3,Fs3]=wavexread('voce_maschile.wav');

lunghezza=25000;

if length(signal1)>lunghezza || length(signal2)>lunghezza ||
length(signal3)>lunghezza;
   lunghezza=max(length(signal1),length(signal2));
   lunghezza=max(lunghezza,length(signal3));
end;

signal1=cat(1,signal1,zeros(lunghezza-length(signal1),1));
signal2=cat(1,signal2,zeros(lunghezza-length(signal2),1));
signal3=cat(1,signal3,zeros(lunghezza-length(signal3),1));

%eseguo la finestratura dei seganli, lo splicing
part1(1:512,1)=signal1(1:length(win),1);
part2(1:512,1)=signal2(1:length(win),1);
part3(1:512,1)=signal3(1:length(win),1);
splice1(1:512,1)=part1(1:512,1)'.*win';
splice2(1:512,1)=part2(1:512,1)'.*win';
splice3(1:512,1)=part3(1:512,1)'.*win';

for i=2:95;

   start=1+(i-1)*(256);
   fin=start+length(win)-1;
   part1(1:512,i)=signal1(start:fin,1);
   part2(1:512,i)=signal2(start:fin,1);
   part3(1:512,i)=signal3(start:fin,1);
   splice1(1:512,i)=part1(1:512,i)'.*win';
   splice2(1:512,i)=part2(1:512,i)'.*win';
   splice3(1:512,i)=part3(1:512,i)'.*win';

end;

%memorizzo le trasformate

for i=1:95;

   fourier1(1:512,i)=fft(splice1(1:512,i));
   fourier2(1:512,i)=fft(splice2(1:512,i));
   fourier3(1:512,i)=fft(splice3(1:512,i));

end;

%moduli per poterli rappresentare
mod_stft1=abs(fourier1);
mod_stft2=abs(fourier2);
mod_stft3=abs(fourier3);
figure(1); surf(mod_stft1);
figure(2); surf(mod_stft2);
figure(3); surf(mod_stft3);


%segnale risintettizzato (utilizzando solamente le stft calcolate)
```

CLUSTERING

APPLICAZIONE MASCHERE BINARIE

IFFT + PhaseUnwrapping --> vedere codici di Antonacci

% [...]

Struttura Dati

lunedì 23 marzo 2009
13.41

Progetto/appunti Roberto

Articoli/Varie

Per il CLUSTERING :

<file://C:\Documents and Settings\Charly\Desktop\Università_Vari\TATI\Libri\Fukunaga.pdf>

<file://C:\Documents and Settings\Charly\Desktop\Università_Vari\TATI\Libri\FeatureExtraction.pdf>

<file://C:\Documents and Settings\Charly\Desktop\Università_Vari\TATI\Libri\Theodoridis.pdf>

<file://C:\Documents and Settings\Charly\Desktop\Università_Vari\TATI\Libri\PRATT-_addison_wesley_-_digital_image_processing_-_3rd_edition.pdf>

Articoli vari di sistemi già realizzati simili al nostro:

<file://C:\Documents and Settings\Charly\Desktop\Università_Vari\TATA\Progetto\ArticoloInternet1.pdf>

Articolo1

domenica 22 marzo 2009
17.35

FrameWor
kGenerale

Inserted from: <file://C:\Documents and Settings\Charly\Desktop\Università_Vari\TATA\Progetto\FrameWorkGenerale.pdf>

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 ($RT_{60} = 128$ ms) conditions. We also investigated the performance under more reverberant conditions ($RT_{60} = 300$ ms). The final section concludes this paper.

2. Binary mask approach to BSS

2.1. Problem description

Suppose that sources $s_1, \ldots, s_N$ are convolutively mixed and observed at M sensors

$$x_j(t) = \sum_{k=1}^{N} \sum_{l} h_{jk}(l) s_k(t-l), \quad j = 1, \ldots, M, \tag{1}$$

where $h_{jk}(l)$ 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 $y_k(t)$ 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 $x_j(t)$ sampled at frequency f_s are converted into frequency-domain time-series signals $x_j(f,t)$ with a T-point short-time Fourier transform (STFT):

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

where $f \in \{0, (1/T)f_s, \ldots, ((T-1)/T)f_s\}$ is a frequency, 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 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 $s_k(t)$ 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 $s_k(t)$. 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.

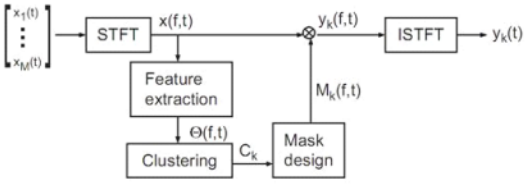


Fig. 1. Basic scheme of binary mask approach.

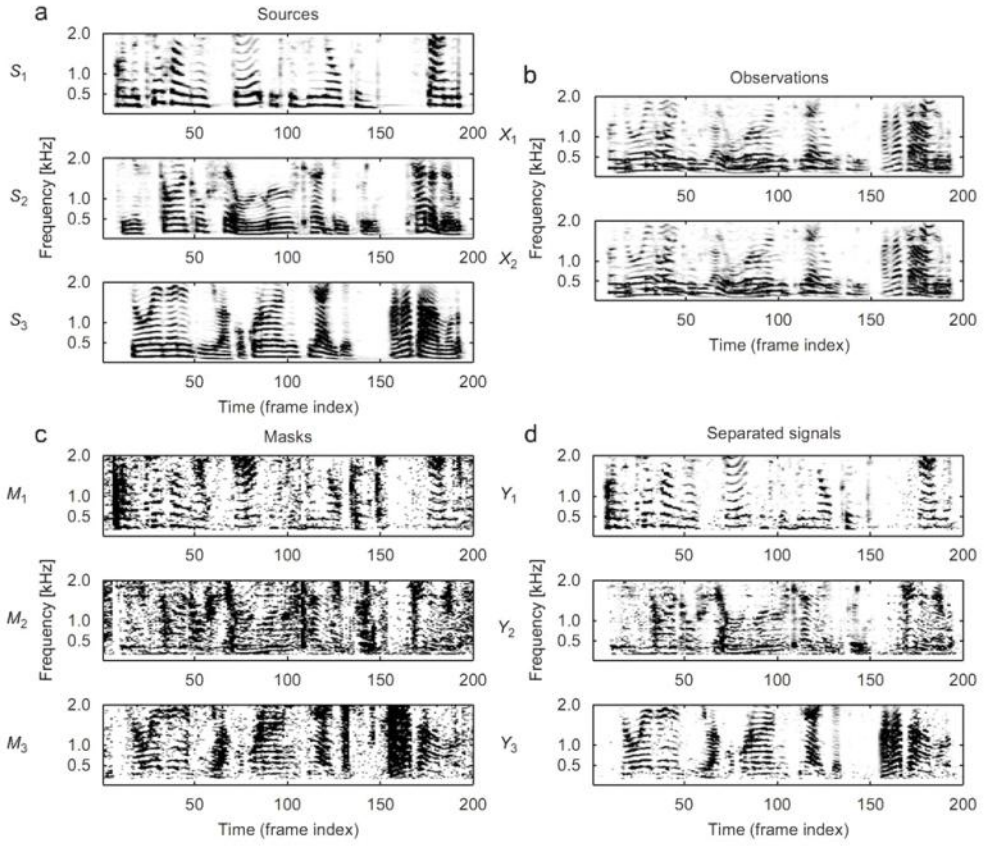


Fig. 2. Example spectra of (a) speech sources, (b) observations, (c) masks and (d) separated signals ($N = 3$, $M = 2$).

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 $\boldsymbol{\Theta}(f,t)$ are calculated by using the frequency–domain observation signals $\mathbf{x}(f,t)$. Here, $\boldsymbol{\Theta}(f,t)$ is a vector that consists of certain geometric features. Generally, the level ratios and phase differences between observations are utilized for $\boldsymbol{\Theta}(f,t)$. Previously employed features are discussed in Section 3, and our newly proposed feature is introduced in Section 4.

Step 3. *Clustering*: Then the features $\boldsymbol{\Theta}(f,t)$ 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 $\mathcal{J}$ of the Euclidean distances (ED) between cluster members and their centroids $\mathbf{c}_k$:

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

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,

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

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

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

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 $\boldsymbol{\Theta}(f,t)$		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)	$\tilde{\Theta}_j(f,t) = \lvert x_j(f,t)\rvert \exp\left[J\frac{\arg[x_j(f,t)/x_J(f,t)]}{\alpha_{jJ}}\right]$ $\boldsymbol{\Theta}(f,t) \leftarrow \tilde{\boldsymbol{\Theta}}(f,t)/\lVert\tilde{\boldsymbol{\Theta}}(f,t)\rVert$		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_{jJ}: 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 $\boldsymbol{\Theta}(f,t)$ 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 $|x_2|/|x_1|$. Fig. 3 shows examples. We can see several large values in the level ratio of (A), although we used omnidirectional microphones where $|x_2|/|x_1| \approx 1$. 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

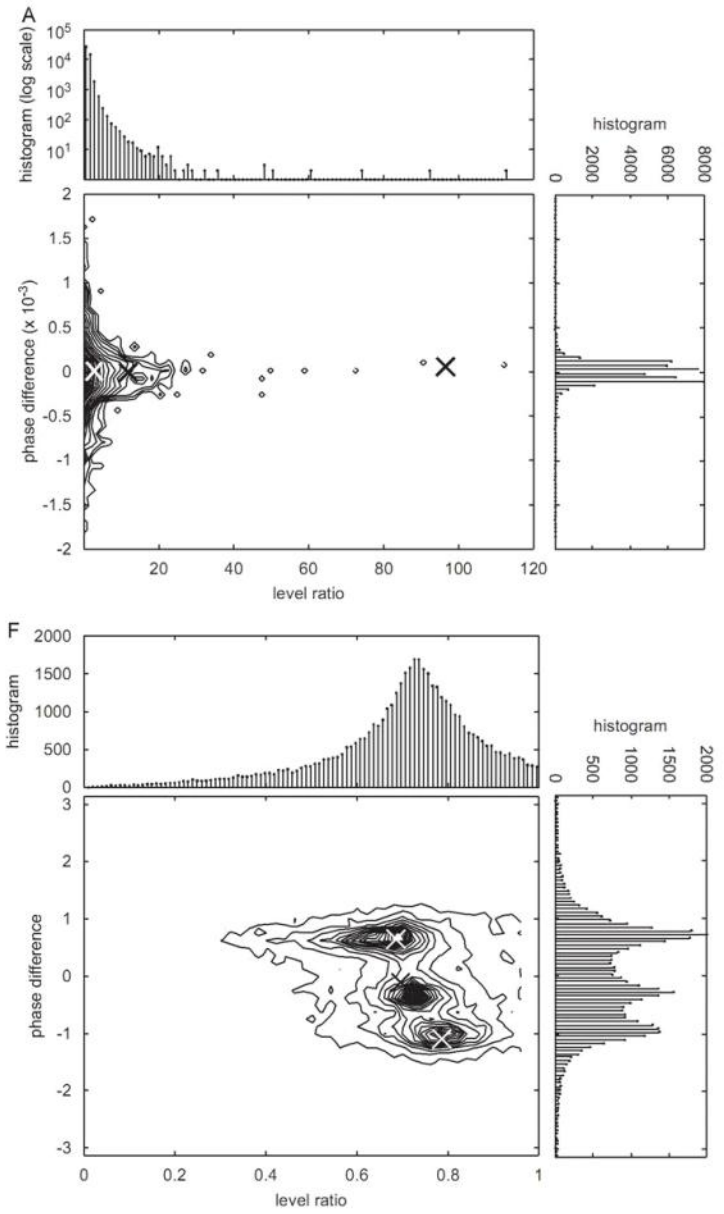


Fig. 3. Example histograms with features (A) and (F). For each feature, top: histogram of the level ratio term of each feature, bottom left: the contour plot of the two-dimensional histogram, bottom right: histogram of the phase difference term of each feature. In the contour plots, × denotes the cluster 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,

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

[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.

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

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

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

where $\tilde{\mathbf{\Theta}}(f,t)=[\tilde{\Theta}_1(f,t),\ldots,\tilde{\Theta}_M(f,t)]^{\mathrm{T}}$. 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

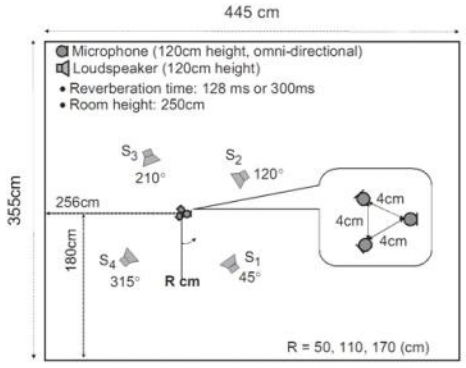


Fig. 4. Room setup ($N = 4, M = 3$).

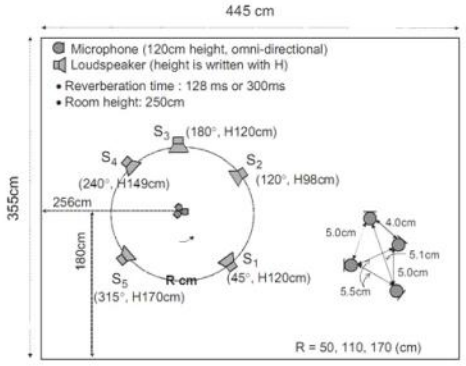


Fig. 5. Room setup ($N = 5, M = 4$).

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 $h_{jk}(f) \approx \exp[-j2\pi f c^{-1}\mathbf{d}_j^{\mathrm{T}}\mathbf{q}_k]$, 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

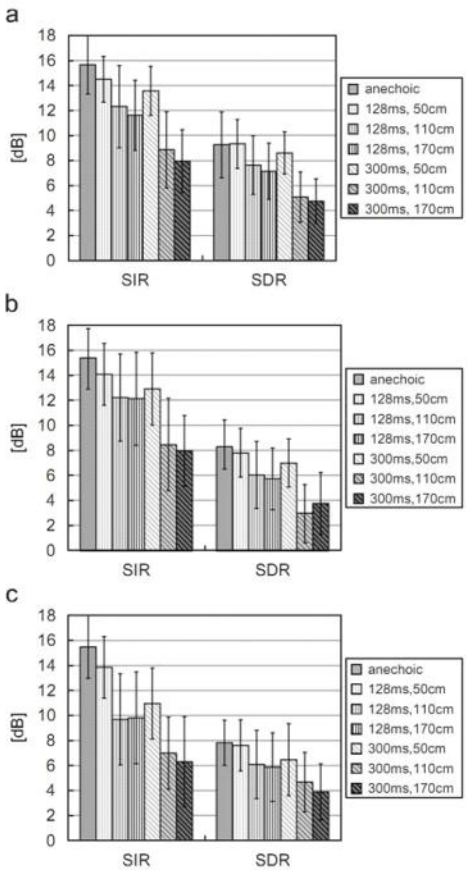


Fig. 6. Average SIR improvement and SDR for each condition. The error bar shows the standard deviation for all outputs and combinations: (a) $N = 4, M = 3$ (average input SIR ≈ -4.8 [dB]); (b) $N = 5, M = 3$ (average input SIR ≈ -6.0 [dB]); (c) $N = 5, M = 4$ (average input SIR ≈ -6.3 [dB]).

$\alpha < 4c^{-1}d_{\max}$, the distance measure (7) for the clustering is not evaluated correctly (see Appendix B), and therefore, the clustering stage failed and the performance worsened.

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

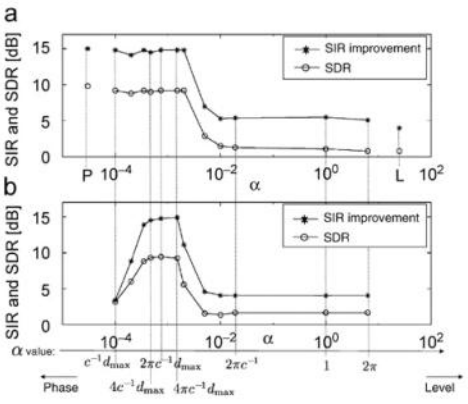


Fig. 7. Relationship between α and separation performance when $N = 4$, $M = 3$, $R = 50$ cm, and $RT_{60} = 128$ ms. (a) with feature (15) and (b) with modified feature (22). In (a), "P" denotes the performance with $\Theta = \Theta^P$, and "L" means with $\Theta = \Theta^L$.

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

Table 3
Average clarity index [dB]

	$R = 50$ cm	$R = 110$ cm	$R = 170$ cm
$RT_{60} = 128$ ms	45.5	40.8	36.6
$RT_{60} = 300$ ms	40.5	34.9	32.7

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.

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

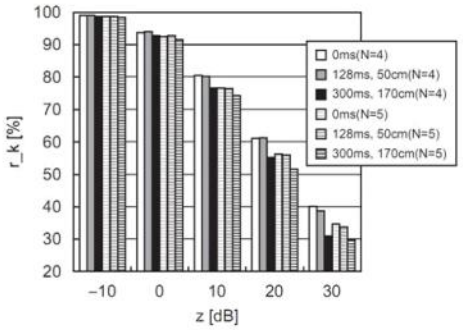


Fig. 8. Source sparseness $r_k(z)$ for some reverberant conditions.

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

Appendix A. Performance measures

The SIR improvement was calculated by

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

where

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

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

where $x_{Jk}(t) = \sum_l h_{Jk}(l)s_k(t-l)$ and $y_{ik}(t)$ is the component of s_k that appears at output $y_i(t)$: $y_i(t) = \sum_{k=1}^{N} y_{ik}(t)$.

The SDR is employed to evaluate the sound quality:

$$\mathrm{SDR}_i = 10\log_{10}\frac{\sum_t |x_{Ji}(t)|^2}{\sum_t |x_{Ji}(t) - \beta y_{ii}(t-D)|^2}(\mathrm{dB}), \quad (A.3)$$

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:

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

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 $l_{jk} - l_{Jk}$:

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

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

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

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

$$|\arg[\Theta]| = \left|\frac{2\pi c^{-1}(l_{jk} - l_{Jk})}{\alpha_j}\right| \leqslant \left|\frac{2\pi c^{-1} d_{\max}}{\alpha_j}\right| \leqslant \frac{\pi}{2}, \quad (B.4)$$

$$\alpha_j \geqslant 4c^{-1}d_{\max}. \quad (B.5)$$

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⟩.

Articolo2

domenica 22 marzo 2009
17.38

SpazioParametri

Inserted from: <file://C:\Documents and Settings\Charly\Desktop\Università_Vari\TATA\Progetto\SpazioParametri.pdf>

Underdetermined Blind Separation of Nondisjoint Sources in the Time-Frequency Domain

Abdeldjalil Aïssa-El-Bey, Nguyen Linh-Trung, Karim Abed-Meraim, *Senior Member, IEEE*, Adel Belouchrani, and Yves Grenier, *Member, IEEE*

***Abstract*—This paper considers the blind separation of nonstationary sources in the underdetermined case, when there are more sources than sensors. A general framework for this problem is to work on sources that are sparse in some signal representation domain. Recently, two methods have been proposed with respect to the time-frequency (TF) domain. The first uses quadratic time-frequency distributions (TFDs) and a clustering approach, and the second uses a linear TFD. Both of these methods assume that the sources are disjoint in the TF domain; i.e., there is, at most, one source present at a point in the TF domain. In this paper, we relax this assumption by allowing the sources to be TF-nondisjoint to a certain extent. In particular, the number of sources present at a point is strictly less than the number of sensors. The separation can still be achieved due to subspace projection that allows us to identify the sources present and to estimate their corresponding TFD values. In particular, we propose two subspace-based algorithms for TF-nondisjoint sources: one uses quadratic TFDs and the other a linear TFD. Another contribution of this paper is a new estimation procedure for the mixing matrix. Finally, then numerical performance of the proposed methods are provided highlighting their performance gain compared to existing ones.**

***Index Terms*—Blind source separation, sparse signal decomposition/representation, spatial time-frequency representation, speech signals, subspace projection, underdetermined/overcomplete representation, vector clustering.**

I. Introduction

SOURCE separation aims at recovering multiple sources from multiple observations (mixtures) received by a set of linear sensors. The problem is said to be "blind" when the observations have been linearly mixed by the transfer medium, while having no *a priori* knowledge of the transfer medium or the sources. Blind source separation (BSS) has applications in several areas, such as communication, speech/audio processing, and biomedical engineering [1]. A fundamental and necessary assumption of BSS is that the sources are statistically independent and thus are often sought solutions using higher order statistical information [2]. If some information about the sources is available at hand, such as temporal coherency [3], source nonstationarity [4], or source cyclostationarity [5], then one can remain in the second-order statistical scenario.

The BSS is said to be *underdetermined* if there are more sources than sensors. In that case, the mixing matrix is not invertible and, consequently, a solution for source estimation must also be found even if the mixing matrix has been estimated. A general framework for underdetermined blind source separation (UBSS) is to exploit the sparseness, if it exists, of the sources in a given signal representation domain [6]. The mixtures are then transformed to this domain; one may then, estimate the transformed sources using their sparseness, and finally recover their time waveforms by source synthesis. For more information on BSS and UBSS methods, see, for example, a recent survey [7].

Recently, several UBSS methods for *nonstationary sources* have been proposed, given that these sources are sparse in the time-frequency (TF) domain [8]–[10]. The first method uses quadratic time-frequency distributions (TFDs), whereas the second one uses a linear TFD. The main assumption used in these methods is that the sources are TF-disjoint; in other words, there is, at most, one source present at any point in the TF domain. This assumption is rather restrictive, though the methods have also showed that they worked well under a quasi-sparseness condition, i.e., sources are TF-almost-disjoint.

In this paper, we want to relax the TF-disjoint condition by allowing the sources to be *nondisjoint* in the TF domain; that is, multiple sources are possibly present at any point in the TF domain. This case has been considered in [11] (which corresponds to part of this paper) and in [12] for the parametric mixing matrix case. In particular, we limit ourselves to the scenario where the number of sources present at any point is smaller than the number of sensors. Under this assumption, the separation of TF-nondisjoint sources is achieved due to *subspace projection*. Subspace projection allows us to identify at any point the sources present, and hence, to estimate the corresponding TFD values of these sources.

The main contribution of this paper is proposing two subspace-based algorithms for UBSS in the TF domain: one uses quadratic TFDs, while the other uses linear TFD. In line with the cluster-based quadratic algorithm proposed in [8], we also propose here a cluster-based algorithm but using a linear TFD, which is not a block-based technique like the quadratic one. Therefore, its low cost computation is useful for processing speech and audio sources. Another contribution of the paper is a method of estimation for the mixing matrix.

The paper is organized as follows. Section II-A formulates the UBSS problem, introduces the underlying TF tools and states some TF conditions necessary for the separation of nonstationary sources in the TF domain. Section III deals

Manuscript received November 7, 2005; revised February 28, 2006. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. A. Rahim Leyman.

A. Aïssa-El-Bey, K. Abed-Meraim, and Y. Grenier are with the Signal and Image Processing Department, École Nationale Supérieure des Télécommunications (ENST) Paris, 75634 Paris, Cedex 13, France (e-mail: elbey@tsi.enst.fr; abed@tsi.enst.fr; grenier@tsi.enst.fr).

N. Linh-Trung is with the College of Technology, Vietnam National University, 144 Xuan Thuy, Cau Giay, Ha Noi, Vietnam (e-mail: linhtrung@ieee.org).

A. Belouchrani is with the École Nationale Polytechnique (ENP), 16200 El Harache, Algiers, Algeria (e-mail: adel.belouchrani@enp.edu.dz).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TSP.2006.888877

with the TF-disjoint sources. It reviews the cluster-based quadratic TF-UBSS algorithm [8] and, from that, proposes a cluster-based linear TF-UBSS algorithm. Section IV proposes two subspace-based TF-UBSS algorithms for TF-nondisjoint sources, using quadratic and linear TFDs. In this section, we propose also a method for the blind estimation of mixing matrix. There is some discussion of the proposed methods in Section V. The performance of the above methods are numerically evaluated in Section VI.

II. Problem Formulation

A. Data Model

Let $s_1(t), \ldots, s_N(t)$ be the desired sources to be recovered from the instantaneous mixtures $x_1(t), \ldots, x_M(t)$ given by

$$\mathbf{x}(t) = \mathbf{A}\mathbf{s}(t) \tag{1}$$

where $\mathbf{s}(t) = [s_1(t), \ldots, s_N(t)]^T$ is the source vector with the superscript ^{T} denoting the transpose operation, $\mathbf{x}(t) = [x_1(t), \ldots, x_M(t)]^T$ is the mixture vector, and $\mathbf{A} = [\mathbf{a}_1, \ldots, \mathbf{a}_N]$ is the mixing matrix of size $M \times N$ that satisfies:

Assumption 1: The column vectors of $\mathbf{A}$ are pairwise linearly independent. That is, for any index pair $i, j \in \mathcal{N}$, where $\mathcal{N} = \{1, \ldots, N\}$, and $i \neq j$, we have $\mathbf{a}_i$ and $\mathbf{a}_j$ linearly independent. This assumption is necessary because if otherwise, we have $\mathbf{a}_1 = \alpha\mathbf{a}_2$ for example, then the input/output relation (1) can be reduced to

$$\mathbf{x}(t) = [\mathbf{a}_1, \mathbf{a}_3, \ldots, \mathbf{a}_N]\,[s_1(t) + \alpha s_2(t), s_3(t), \ldots, s_N(t)]^T$$

and hence the separation of $s_1(t)$ and $s_2(t)$ is inherently impossible.

It is known that BSS is only possible up to some scaling and permutation. We take advantage of these indeterminacies to further assume, without loss of generality, that the column vectors of $\mathbf{A}$ all have unit norm, i.e., $\|\mathbf{a}_i\| = 1$ for all $i \in \mathcal{N}$.

The sources are nonstationary, that is their frequency spectra vary in time. Often, nonstationarity imposes more difficulties on a problem; however, in this case, it actually offers a solution: one can solve the BSS problem without using higher order approaches by directly exploiting the additional information of this TF diversity across the spectra; this solution was proposed in [4]. We defer to a little later making TF assumptions on the sources, and for now we introduce the concept of TF signal processing.

B. Time-Frequency Distributions

TF signal processing provides effective tools for analyzing nonstationary signals, whose frequency content varies in time. This concept is a natural extension of both the time domain and the frequency domain processing that involve representing signals in a two-dimensional (2-D) space the joint TF domain, hence providing a distribution of signal energy versus time and frequency simultaneously. For this reason, a TF representation is commonly referred to as a TFD.

The general class of quadratic TFDs of an analytic signal $z(t)$ is defined as [13]

$$\rho_{zz}(t,f) \triangleq \int\int\int_{-\infty}^{\infty} e^{j2\pi\nu(u-t)}\Gamma(\nu,\tau) \times z\left(u+\frac{\tau}{2}\right) z^*\left(u-\frac{\tau}{2}\right) e^{-j2\pi f\tau}\, d\nu du d\tau \tag{2}$$

where $\Gamma(\nu,\tau)$ is a 2-D function in the so-called ambiguity domain and is called the Doppler-lag kernel, and the superscript $(^*)$ denotes the conjugate operator. We can design a TFD with certain desired properties by properly constraining Γ.

When $\Gamma(\nu,\tau) = 1$ we have the following famous Wigner–Ville distribution (WVD):

$$\rho_{zz}^{\mathrm{wvd}}(t,f) \triangleq \int_{-\infty}^{\infty} z\left(t+\frac{\tau}{2}\right) z^*\left(t-\frac{\tau}{2}\right) e^{-j2\pi f\tau}\, d\tau. \tag{3}$$

The WVD is the most widely studied TFD. It achieves maximum energy concentration in the TF plane around the instantaneous frequency for linear frequency-modulated (LFM) signals. However, it is in general nonpositive, and it introduces the so-called "cross-terms" when multiple frequency laws (e.g., two LFM components) exist in the signals, due to the quadratic multiplication of shifted versions of the signals.

Another well-known TFD and most used in practice is the short-time Fourier transform (STFT)

$$\mathcal{S}_z(t,f) \triangleq \int_{-\infty}^{\infty} z(\tau)h(\tau-t)\, e^{-j2\pi f\tau} d\tau \tag{4}$$

where $h(t)$ is a window function. Note that the STFT is a *linear* TFD,[1] and its quadratic version, called the spectrogram (SPEC), is defined as

$$\rho_{zz}^{\mathrm{spec}}(t,f) \triangleq |\mathcal{S}_z(t,f)|^2. \tag{5}$$

Clearly, from the definition, there is no cross-terms effect present in STFT, hence in the SPEC. However, these distributions have very low TF resolution in comparison with the WVD. The low cost of implementation for the STFT, hence for the SPEC, in comparison with that for the WVD and, together with the advantage of being free of cross terms, justifies the fact that the STFT is most used in practice, especially for speech or audio signals. However, when it comes to frequency-modulated (FM) signals, the WVD is preferred.

To combine the high resolution of the WVD while using the free cross-term property of the SPEC, the masked Wigner–Ville distribution (MWVD) is derived so that

$$\rho_{zz}^{\mathrm{mwvd}}(t,f) \triangleq \rho_{zz}^{\mathrm{wvd}}(t,f) \cdot \rho_{zz}^{\mathrm{spec}}(t,f). \tag{6}$$

There are many other useful TFDs in the literature, notably those that give high TF resolution while effectively minimizing the cross terms, for example, the B distribution [14]. However, we only introduce here the TFDs above since they will be used in the later sections.

[1]In fact, the STFT does not represent an energy distribution of the signal in the TF plane. However, for simplicity, we still refer to it as a TFD.

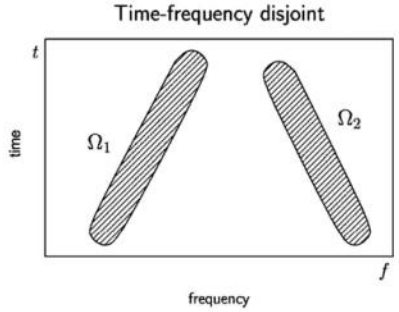


Fig. 1. Source TF-disjoint condition: $\Omega_1 \cap \Omega_2 = \emptyset$ (when $\Omega_1 \cap \Omega_2 \approx \emptyset$, sources are said to be TF-almost-disjoint).

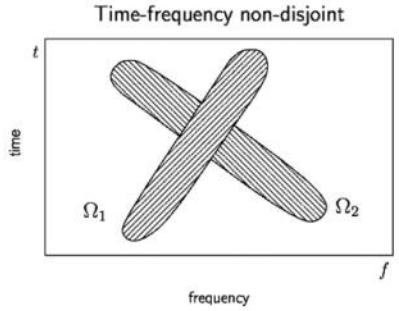


Fig. 2. TF-nondisjoint condition: $\Omega_1 \cap \Omega_2 \neq \emptyset$.

C. TF Conditions on Sources

Now, as we have introduced the concept of TF signal processing as a useful tool for analyzing nonstationary signals, some TF conditions need to be applied to the sources. Note that the TF method in [4] does not work for UBSS because the mixing matrix is not invertible. In order to deal with UBSS, one often seeks for a sparse representation of the sources [6]. In other words, if the sources can be sparsely represented in some domain, then the separation is to be carried out in that domain to exploit the sparseness.

1) TF-Disjoint Sources: Recently, there have been several UBSS methods, notably those in [8] and [9], in which the TF domain has been chosen to be the underlaying sparse domain. These two papers have based their solutions on the assumption that the sources are disjoint in the TF domain. Mathematically, if Ω_1 and Ω_2 are the TF supports of two sources $s_1(t)$ and $s_2(t)$, then $\Omega_1 \cap \Omega_2 = \emptyset$. This condition can be illustrated in Fig. 1. However, this is a rather strict assumption. A more practical assumption is that the sources are almost-disjoint in the TF domain [8], allowing some small overlapping in the TF domain, for which the above two methods also worked.

2) TF-Nondisjoint Sources: In this paper, we want to relax the TF-disjoint condition by allowing the sources to be nondisjoint in the TF domain, as illustrated in Fig. 2.

This is motivated by a drawback of the method in [8]. Although this method worked well under the TF-almost-disjoint condition, it did not explicitly treat the TF regions where the sources were allowed to have some small overlapping. A point at the overlapping of two sources was assigned "by chance" to belong to only one of the sources. As a result, the source that picks up this point will have some information of the other source while the latter loses some information of its own. The loss of information can be recovered to some extent by the interpolation at the intersection point using TF synthesis. However, for the other source, there is an interference at this point, hence the separation performance may degrade if no treatment is provided. If the number of overlapping points increases (i.e., the TF-almost-disjoint condition is violated), the performance of the separation is expected to degrade unless the overlapping points are treated.

This paper will give such a treatment using subspace projection. Therefore, we will allow the sources to be *nondisjoint* in the TF domain; that is, multiple sources are allowed to be present at any point in the TF domain. However, instead of being inevitably nondisjoint, we limit ourselves by making the following constraint.

Assumption 2: The number of sources that contribute their energy at any TF point is strictly less than the number of sensors.

In other words, for the configuration of M sensors, there exist at most $(M-1)$ sources at any point in the TF domain. For the special case when $M = 2$, Assumption 2 reduces to the disjoint condition.

We also make another assumption on the TF conditioning of the sources.

Assumption 3: For each source, there exists a region in the TF domain, where this source exists alone.

Note that, this assumption is easily met and hence not restrictive for audio sources and FM-like signals. Also, it should be noted that this last assumption is, however, not a restriction on the use of subspace projection, because it will only be used later for the estimation of the mixing matrix. If otherwise, the mixing matrix can be obtained by another method, for example the one in [15], then Assumption 3 can be omitted.

III. Cluster-Based TF-UBSS Approach for Disjoint Sources

A. Quadratic TFD Approach

In this section, we review a method proposed in [8] based on the idea of clustering; hence, it is now referred to as the *cluster-based quadratic TF-UBSS algorithm*. For a signal vector $\mathbf{z}(t) = [z_1(t), \ldots, z_N(t)]^T$, the STFD matrix is given by [4]

$$\mathbf{D}_{\mathbf{zz}}(t,f) \triangleq \begin{bmatrix} \rho_{z_1 z_1}(t,f) & \cdots & \rho_{z_1 z_N}(t,f) \\ \vdots & \ddots & \vdots \\ \rho_{z_N z_1}(t,f) & \cdots & \rho_{z_N z_N}(t,f) \end{bmatrix} \quad (7)$$

where, for $i, j \in \mathcal{N}$, $\rho_{z_i z_j}(t,f)$ is the quadratic cross-TFD between $z_i(t)$ and $z_j(t)$ as obtained by (2), but with the first z being replaced by z_i and the second by z_j. By definition, the STFD takes into account the spatial diversity.

By applying the STFD defined in (7) on both sides of the BSS model in (1), we obtain the following TF-transformed structure:

$$\mathbf{D}_{\mathbf{xx}}(t,f) = \mathbf{A}\mathbf{D}_{\mathbf{ss}}(t,f)\mathbf{A}^H \quad (8)$$

TABLE I
CLUSTER-BASED QUADRATIC TF-UBSS ALGORITHM USING STFD

1) Mixture STFD computation by (10); noise thresholding by (11).
2) Noise thresholding and auto-source point selection by (11).
3) Vector clustering by (12) and k-means algorithm; source TFD estimation by (13).
4) Source TF synthesis by [16].

where $\mathbf{D_{ss}}(t,f)$ and $\mathbf{D_{xx}}(t,f)$ are, respectively, the source STFD matrix and mixture STFD matrix.

Let us call an *autosource TF point* a point at which there is a true energy contribution/concentration of source or sources in the TF domain, and a *cross-source point* a point at which there is a "false" energy contribution (due to the cross-term effect of quadratic TFDs). Note that, at other points with no energy contribution, the TFD value is ideally equal to zero. Under the assumption that all sources are disjoint in the TF domain, there is only one source present at any autosource point. Therefore, the structure of $\mathbf{D_{xx}}(t,f)$ is reduced to

$$\mathbf{D_{xx}}(t_a,f_a) = \rho_{s_i s_i}(t_a,f_a)\mathbf{a}_i\mathbf{a}_i^H, \forall (t_a,f_a) \in \Omega_i, \quad (9)$$

where Ω_i denotes, hereafter, the TF support of source $s_i(t)$.

The observation (9) suggests that for all $(t_a,f_a) \in \Omega_i$, the corresponding set of STFD matrices $\{\mathbf{D_{xx}}(t_a,f_a)\}$ will have the same principal eigenvector $\mathbf{a}_i$. It is this observation that leads to the general separation method using quadratic TFDs in [8]. Indeed, [8] proposed several algorithms and pointed out that the choice of the TFD should be made carefully in order to have a "clean" (cross-term-free) TFD representation of the mixture and chose the MWVD as a good candidate. This algorithm is summarized in Table I and further detailed below for later use.

1) STFD Mixture Computation and Noise Thresholding: The STFD of the mixtures using the MWVD is computed by the following:

$$\left[\mathbf{D}_{\mathbf{xx}}^{\mathrm{wvd}}(t,f)\right]_{k,l} = \rho_{x_k x_l}^{\mathrm{wvd}}(t,f) \quad (10a)$$

$$\left[\mathbf{D}_{\mathbf{xx}}^{\mathrm{stft}}(t,f)\right]_{k,l} = \begin{cases} \mathcal{S}_{x_k}(t,f), & \text{for } k=l, \\ 0, & \text{otherwise} \end{cases} \quad (10b)$$

$$\mathbf{D}_{\mathbf{xx}}^{\mathrm{mwvd}}(t,f) = \mathbf{D}_{\mathbf{xx}}^{\mathrm{wvd}}(t,f) \odot \left|\mathbf{D}_{\mathbf{xx}}^{\mathrm{stft}}(t,f)\right|^2. \quad (10c)$$

In (10), $k,l \in \mathcal{N}$, and $\odot$ denotes the Hadamard product.

2) Noise Thresholding and Autosource Point Selection: A "noise thresholding" procedure is used to keep only those points having sufficient energy, i.e., autosource points. One way to do this is as follows: for each time-slice (t_p,f) of the TFD representation, apply the following criterion for all the frequency points f_q belonging to this time-slice:

$$\text{If} \quad \frac{\left\|\mathbf{D}_{\mathbf{xx}}^{\mathrm{mwvd}}(t_p,f_q)\right\|}{\max_f\left\{\left\|\mathbf{D}_{\mathbf{xx}}^{\mathrm{mwvd}}(t_p,f)\right\|\right\}} > \epsilon_1, \quad \text{keep}(t_p,f_q) \quad (11)$$

where ϵ_1 is a small threshold (typically, $\epsilon_1 = 0.05$). This "hard thresholding" procedure has been preferred to the "soft thresholding" using power-weighting of [9] as it contributes also to reducing the computation complexity. The set of all the autosource points is denoted by Ω. Since sources are TF-disjoint, we have $\Omega = \bigcup_{i=1}^{N} \Omega_i$. This partition is found in the following way.

3) Vector Clustering and Source TFD Estimation: For each point $(t_a,f_a) \in \Omega$, compute its corresponding spatial direction $\mathbf{a}(t_a,f_a)$

$$\mathbf{a}(t_a,f_a) = \frac{\mathrm{diag}\left\{\underline{\mathbf{D}}_{\mathbf{xx}}^{\mathrm{stft}}(t_a,f_a)\right\}}{\left\|\mathrm{diag}\left\{\underline{\mathbf{D}}_{\mathbf{xx}}^{\mathrm{stft}}(t_a,f_a)\right\}\right\|} \quad (12)$$

and force it, without loss of generality, to have the first entry real and positive.

Having the set of spatial direction $\{\mathbf{a}(t_a,f_a) \,|\, (t_a,f_a) \in \Omega\}$, one can cluster them into N classes using any unsupervised clustering algorithm (see [17] for different clustering methods). The clustering algorithm used in [8] is rather sensitive due to the threshold in use; a robust method should be investigated, and this deserves another contribution. If the number of sources has been well estimated, one can use the so-called k-means clustering algorithm [17] to achieve a good clustering performance. The output of the clustering algorithm is a set of N classes $\{C_i \,|\, i \in \mathcal{N}\}$. Also, the collection of all the points that correspond to all the vectors in the class C_i forms the TF support Ω_i of the source $s_i(t)$.

Then, one can estimate the TFD of the source $s_i(t)$ (up to a scalar constant) as

$$\hat{\rho}_{s_i}^{\mathrm{wvd}}(t,f) = \begin{cases} \mathrm{trace}\left\{\underline{\mathbf{D}}_{\mathbf{xx}}^{\mathrm{wvd}}(t,f)\right\}, & (t,f) \in \Omega_i, \\ 0, & \text{otherwise} \end{cases}. \quad (13)$$

4) Source TF Synthesis: Having obtained the source TFD estimate $\hat{\rho}_{s_i}^{\mathrm{wvd}}(t,f)$, the estimation of the source $s_i(t)$ can be done through a TF synthesis algorithm. The method in [16] is used for TF synthesis from a WVD estimate, based on the following inversion property of the WVD [13]:

$$x(t) = \frac{1}{x^*(0)} \int_{-\infty}^{\infty} \rho_x^{\mathrm{wvd}}\left(\frac{t}{2},f\right) e^{j2\pi ft} df$$

which implies that the signal can be reconstructed to within a complex exponential constant $e^{j\alpha} = x^*(0)/|x(0)|$ given $|x(0)| \neq 0$.

It can be observed that in this version of the quadratic TF-UBSS algorithm, the STFD matrices are not fully needed as only their diagonal entries are used in the algorithm. This should be taken into account to reduce the computational cost.

B. Linear TFD Approach

As we have seen before, the STFT is often used for speech/audio signals because of its low computational cost. Therefore, in this section, we briefly review the STFT method in [9] and propose simultaneously a *cluster-based linear TF-UBSS algorithm* using the STFT to avoid some of the drawbacks in [9].

TABLE II
CLUSTER-BASED LINEAR TF-UBSS ALGORITHM USING STFT

1) Mixture STFT computation by (17); noise thresholding by (18)
2) Vector clustering by (19) and (20).
3) Source STFT estimation by (21).
4) Source TF synthesis by [18].

First, under the transformation into the TF domain using the STFT, the model in (1) becomes

$$\mathcal{S}_{\mathbf{x}}(t,f) = \mathbf{A}\mathcal{S}_{\mathbf{s}}(t,f), \tag{14}$$

where $\mathcal{S}_{\mathbf{x}}(t,f)$ is the mixture STFT vector and $\mathcal{S}_{\mathbf{s}}(t,f)$ is the source STFT vector. Under the assumption that all sources are disjoint in the TF domain, (14) is reduced to

$$\mathcal{S}_{\mathbf{x}}(t_a,f_a) = \mathbf{a}_i\mathcal{S}_{s_i}(t_a,f_a), \quad \forall (t_a,f_a) \in \Omega_i, \forall i \in \mathcal{N}. \tag{15}$$

Now, in [9], the structure of the mixing matrix is particular in that it has only two rows (i.e., the method uses only two sensors) and the first row of the mixing matrix contains all 1's. Then, (15) is expanded to

$$\begin{bmatrix} \mathcal{S}_{x_1}(t_a,f_a) \\ \mathcal{S}_{x_2}(t_a,f_a) \end{bmatrix} = \begin{bmatrix} 1 \\ a_{2,i} \end{bmatrix} \mathcal{S}_{s_i}(t_a,f_a)$$

which results in

$$a_{2,i} = \frac{\mathcal{S}_{x_2}(t_a,f_a)}{\mathcal{S}_{x_1}(t_a,f_a)}. \tag{16}$$

Therefore, all the points for which the ratios on the right-hand side of (16) have the same value form the TF support Ω_i of a single source, say $s_i(t)$. Then, the STFT estimate of $s_i(t)$ is computed by

$$\hat{\mathcal{S}}_{s_i}(t,f) = \begin{cases} \mathcal{S}_{x_1}(t,f), & \forall (t,f) \in \Omega_i \\ 0, & \text{otherwise} \end{cases}.$$

The source estimate $\hat{s}_i(t)$ is then obtained by converting $\hat{\mathcal{S}}_{s_i}(t,f)$ to the time domain using inverse STFT [18]. Note that, the extension of the UBSS method in [9] to more than two sensors is a difficult task. Second, the division on the right-hand side of (16) is prone to error if the denominator is close to zero.

To avoid the above-mentioned problems, we propose here a modified version of the previous method referred to as the cluster-based linear TF-UBSS algorithm. In particular, from the observation (15), we can deduce the separation algorithm as shown next, and summarized in Table II.

1) Mixture STFT Computation and Noise Thresholding: Compute the STFT of the mixtures, $\mathcal{S}_{\mathbf{x}}(t,f)$, by applying (4) for each of the mixture in $\mathbf{x}(t)$, as follows:

$$\mathcal{S}_{x_i}(t,f) = \int_{-\infty}^{\infty} x_i(\tau)h(\tau - t)e^{-j2\pi f\tau}d\tau, \quad i = 1,\ldots,M \tag{17a}$$

$$\mathcal{S}_{\mathbf{x}}(t,f) = [\mathcal{S}_{x_1}(t,f),\ldots,\mathcal{S}_{x_M}(t,f)]^{\mathcal{T}}. \tag{17b}$$

Since the STFT is totally free of cross terms, a point with a nonzero TFD value is ideally an autosource point. Practically, we can select all autosource points by only applying a noise thresholding procedure as that in the cluster-based quadratic TF-UBSS algorithm. In particular, for each time-slice (t_p,f) of the TFD representation, apply the following criterion for all the frequency points f_k belonging to this time-slice:

$$\text{If} \quad \frac{\|\mathcal{S}_{\mathbf{x}}(t_p,f_k)\|}{\max_f\{\|\mathcal{S}_{\mathbf{x}}(t_p,f)\|\}} > \epsilon_2, \quad \text{then keep } (t_p,f_k) \tag{18}$$

where ϵ_2 is a small threshold (typically, $\epsilon_2 = 0.05$). Then, the set of all selected points Ω is expressed by $\Omega = \bigcup_{i=1}^{N}\Omega_i$, where Ω_i is the TF support of the source $s_i(t)$. Note that the effects of spreading the noise energy while localizing the source energy in the time-frequency domain amounts to increasing the robustness of the proposed method with respect to noise. Hence, by (18) (or (11)), we would keep only time-frequency points where the signal energy is significant; the other time-frequency points are rejected, i.e., not further processed, since they are considered to represent noise contribution only. Also, due to the noise energy spreading, the contribution of the noise in the source time-frequency points is relatively, negligible at least for moderate and high signal-to-noise ratios (SNRs).

2) Vector Clustering and Source TFD Estimation: The clustering procedure can be done in a similar manner as in the quadratic algorithm. First, we obtain the spatial direction vectors by

$$\mathbf{v}(t_a,f_a) = \frac{\mathcal{S}_{\mathbf{x}}(t_a,f_a)}{\|\mathcal{S}_{\mathbf{x}}(t_a,f_a)\|}, \quad (t_a,f_a) \in \Omega \tag{19}$$

and force them, without loss of generality, to have the first entry real and positive.

Next, we cluster these vectors into N classes $\{C_i \,|\, i \in \mathcal{N}\}$, using the k-means clustering algorithm. The collection of all points, whose vectors belong to the class C_i, now forms the TF support Ω_i of the source $s_i(t)$. Then, the column vector $\mathbf{a}_i$ of $\mathbf{A}$ is estimated as the centroid of this set of vectors

$$\hat{\mathbf{a}}_i = \frac{1}{|C_i|}\sum_{(t,f)\in\Omega_i}\mathbf{v}(t,f) \tag{20}$$

where $|C_i|$ is the number of vectors in this class.

Therefore, we can estimate the STFT of each source $s_i(t)$ by

$$\hat{\mathcal{S}}_{s_i}(t,f) = \begin{cases} \hat{\mathbf{a}}_i^H\mathcal{S}_{\mathbf{x}}(t,f), & \forall (t,f) \in \Omega_i \\ 0, & \text{otherwise} \end{cases} \tag{21}$$

since, from (15), we have

$$\hat{\mathbf{a}}_i^H\mathcal{S}_{\mathbf{x}}(t,f) = \hat{\mathbf{a}}_i^H\mathbf{a}_i\mathcal{S}_{s_i}(t,f) \approx \mathcal{S}_{s_i}(t,f), \quad \forall (t,f) \in \Omega_i.$$

Note that the STFT is a particular form of wavelet transforms which have been used in [19] for the UBSS of image signals.

IV. SUBSPACE-BASED TF-UBSS APPROACH FOR NONDISJOINT SOURCES

We have seen the cluster-based TF-UBSS methods, using either quadratic TFDs such as the MWVD or linear TFDs such as the STFT, as summarized in Table I or Table II, respectively. These methods relied on the assumption that the sources were TF-disjoint, which has led to the enabling TF-transformed structures in (9) or (15). When the sources are nondisjoint in the TF domain, then these equations are no longer true.

TABLE III
SUBSPACE-BASED QUADRATIC TF-UBSS ALGORITHM USING MWVD

1) Mixture STFD computation by (10).
2) Noise thresholding and auto-source point selection by (11).
3) Single-source auto-source point selection by (29); mixing ma- trix estimation by (30) and (31)
4) For all auto-source points, perform subspace-based TFD esti- mation of sources by (25), (27) and (28)
5) Source TF synthesis by [16].

Under the TF-nondisjoint condition, stated in Assumption 2, we propose in this section two alternative methods: one for quadratic TFDs and the other for linear TFDs, for the UBSS problem using subspace projection.

A. Subspace-Based Quadratic TF-UBSS Algorithm

Recall that the first two steps of the cluster-based quadratic TF-UBSS algorithm do not rely on the assumption of TF-disjoint sources (see Table I). Therefore, we can reuse these steps to obtain the set of autosource points Ω. Now, under the TF-nondisjoint condition, consider an autosource point $(t_b, f_b) \in \Omega$ such that there are K sources, $K < M$, present at this point. Our goal is to identify the sources present at (t_b, f_b) and to estimate the energy each of these sources contributes.

Denote $\alpha_1, \ldots, \alpha_K \in \mathcal{N}$ the indexes of the sources present at (t_b, f_b), and define the following:

$$\tilde{\mathbf{s}} = [s_{\alpha_1}(t), \ldots, s_{\alpha_K}(t)]^T \tag{22a}$$

$$\tilde{\mathbf{A}} = [\mathbf{a}_{\alpha_1}, \ldots, \mathbf{a}_{\alpha_K}]. \tag{22b}$$

Then, under Assumption 2, (8) is reduced to

$$\mathbf{D}_{\mathbf{xx}}^{\mathrm{wvd}}(t_b, f_b) = \tilde{\mathbf{A}}\mathbf{D}_{\tilde{\mathbf{s}}\tilde{\mathbf{s}}}(t_b, f_b)\tilde{\mathbf{A}}^H. \tag{23}$$

Consequently, given that $\mathbf{D}_{\tilde{\mathbf{s}}\tilde{\mathbf{s}}}$ is of full rank, we have

$$\mathrm{Range}\{\mathbf{D}_{\mathbf{xx}}(t_b, f_b)\} = \mathrm{Range}\{\tilde{\mathbf{A}}\}. \tag{24}$$

Let $\mathbf{P}$ be the orthogonal projection matrix onto the noise subspace of $\mathbf{D}_{\mathbf{xx}}^{\mathrm{wvd}}(t_b, f_b)$. Then, from (24), we obtain

$$\mathbf{P} = \mathbf{I} - \mathbf{V}\mathbf{V}^H \tag{25}$$

and

$$\begin{cases} \mathbf{P}\mathbf{a}_i = \mathbf{0}, & \forall i \in \{\alpha_1, \ldots, \alpha_K\} \\ \mathbf{P}\mathbf{a}_i \neq \mathbf{0}, & \forall i \in \mathcal{N} \backslash \{\alpha_1, \ldots, \alpha_K\} \end{cases}. \tag{26}$$

In (25), $\mathbf{V}$ is the matrix formed by the K principal singular eigenvectors of $\mathbf{D}_{\mathbf{xx}}(t_b, f_b)$.

Assuming that $\mathbf{A}$ has been estimated by some method, the observation in (26) enables us to identify the indexes $\alpha_1, \ldots, \alpha_K$, and hence, the sources present at (t_b, f_b). In practice, to take into account the estimation noise, one can detect these indexes by detecting the K smallest values from the set $\{\|\mathbf{P}\mathbf{a}_i\|\} | i \in \mathcal{N}\}$, as mathematically expressed by

$$\{\alpha_1, \ldots, \alpha_K\} = \arg\min^K \{\|\mathbf{P}\mathbf{a}_i\| \, | i \in \mathcal{N}\} \tag{27}$$

where $\min^K$ denotes the minimization to obtain the K smallest values. The TFD values of the K sources at (t_b, f_b) are estimated as the diagonal elements of the following matrix:

$$\hat{\mathbf{D}}_{\tilde{\mathbf{s}}\tilde{\mathbf{s}}}(t_b, f_b) \approx \tilde{\mathbf{A}}^{\#}\mathbf{D}_{\mathbf{xx}}(t_b, f_b)(\tilde{\mathbf{A}}^{\#})^H \tag{28}$$

where the superscript ($^{\#}$) is the Moore–Penrose's pseudoinversion operator.

Here, we propose also an estimation method for $\mathbf{A}$ by using Assumption 3. This assumption states that, for each source $s_i(t)$, there exists a TF region $\mathcal{R}_i$ where $s_i(t)$ exists alone. In other words, $\mathcal{R}_i$ contains all the single-source autosource points of $s_i(t)$. Therefore, we can reuse the observation (9) in the TF-disjoint case, but for some TF regions, as follows:

$$\mathbf{D}_{\mathbf{xx}}(t, f) = \rho_{s_i s_i}(t, f)\mathbf{a}_i\mathbf{a}_i^H, \quad \forall (t, f) \in \mathcal{R}_i, \forall i \in \mathcal{N}.$$

The union of these regions $\mathcal{R} = \bigcup_{i=1}^{N} R_i$ is detected by the following:

$$\text{If} \left| \frac{\lambda_{\max}\{\mathbf{D}_{\mathbf{xx}}^{\mathrm{wvd}}(t, f)\}}{\mathrm{trace}\{\mathbf{D}_{\mathbf{xx}}^{\mathrm{wvd}}(t, f)\}} - 1 \right| < \epsilon_3, \quad \text{then } (t, f) \in \mathcal{R} \tag{29}$$

where ϵ_3 is a small threshold value (typically, $\epsilon_3 \leq 0.1$) and $\lambda_{\max}\{\mathbf{D}_{\mathbf{xx}}^{\mathrm{wvd}}(t, f)\}$ denotes the maximum eigenvalue of $\mathbf{D}_{\mathbf{xx}}^{\mathrm{wvd}}(t, f)$. Then, we can apply the same vector clustering procedure as in Section III-A-3) to estimate $\mathbf{A}$. In particular, we first obtain all the spatial direction vectors

$$\mathbf{a}(t, f) = \frac{\mathrm{diag}\left\{\underline{\mathbf{D}}_{\mathbf{xx}}^{\mathrm{stft}}(t, f)\right\}}{\left\|\mathrm{diag}\left\{\underline{\mathbf{D}}_{\mathbf{xx}}^{\mathrm{stft}}(t, f)\right\}\right\|}, \quad \forall (t, f) \in \mathcal{R}. \tag{30}$$

Next, we cluster these vectors into N classes $\{D_i \, | \, i \in \mathcal{N}\}$ using the k-means clustering algorithm. The collection of all points, whose vectors belong to the class D_i, now forms the TF region $\mathcal{R}_i$ of the source $s_i(t)$. Finally, the column vectors $\mathbf{A}$ are estimated as the centroid vectors of these classes as

$$\hat{\mathbf{a}}_i = \frac{1}{|D_i|} \sum_{(t,f) \subset \mathcal{R}_i} \mathbf{a}(t, f), \quad \forall i \in \mathcal{N} \tag{31}$$

where D_i is the number of points in $\mathcal{R}_i$.

Table III gives a summary of the subspace-based quadratic TF-UBSS algorithm.

B. Subspace-Based Linear TF-UBSS Algorithm

Similarly, we propose here a subspace-based linear TF-UBSS algorithm for TF-nondisjoint sources using STFT. We also use the first step of the cluster-based linear TF-UBSS algorithm (see Table II) to obtain all the autosource points Ω. Under

TABLE IV
SUBSPACE-BASED LINEAR TF-UBSS ALGORITHM USING STFT

1) STFT computation.
2) Noise thresholding and auto-source point selection
3) Mixing matrix estimation by (20) and (37), and k-means algorithm.
4) For all auto-source points, perform subspace-based TFD estimation of sources by (33), (35) and (36).
5) Source TF synthesis by [18].

the TF-nondisjoint condition, consider an autosource point $(t_b, f_b) \in \Omega$ at which there are K sources $s_{\alpha_1}(t), \ldots, s_{\alpha_K}(t)$ present, with $K < M$. Then, (8) is reduced to the following:

$$\mathcal{S}_{\mathbf{x}}(t_b, f_b) = \tilde{\mathbf{A}}\mathcal{S}_{\tilde{\mathbf{s}}}(t_b, f_b), \quad \forall (t_b, f_b) \in \Omega \tag{32}$$

where $\tilde{\mathbf{A}}$ and $\tilde{\mathbf{s}}$ are as previously defined in (22).

Let $\mathbf{Q}$ represent the orthogonal projection matrix onto the noise subspace of $\tilde{\mathbf{A}}$. Then, $\mathbf{Q}$ can be computed by

$$\mathbf{Q} = \mathbf{I} - \tilde{\mathbf{A}}(\tilde{\mathbf{A}}^H\tilde{\mathbf{A}})^{-1}\tilde{\mathbf{A}}^H. \tag{33}$$

We have the following observation:

$$\begin{cases} \mathbf{Q}\mathbf{a}_i = 0, & i \in \{\alpha_1, \ldots, \alpha_K\} \\ \mathbf{Q}\mathbf{a}_i \neq 0, & i \in \mathcal{N}\backslash\{\alpha_1, \ldots, \alpha_K\} \end{cases}. \tag{34}$$

If $\mathbf{A}$ has already been estimated by some method, then this observation gives us the criterion to detect the indexes $\alpha_1, \ldots, \alpha_K$; and hence, the contributing sources at the autosource point (t_b, f_b). In practice, to take into account noise, one detects the column vectors of $\tilde{\mathbf{A}}$, minimizing

$$\{\alpha_1, \ldots, \alpha_K\} = \arg \min_{\beta_1, \ldots, \beta_K} \left\{ \|\mathbf{Q}\mathcal{S}_{\mathbf{x}}(t, f)\| |\tilde{\mathbf{A}}_\beta \right\} \tag{35}$$

where $\tilde{\mathbf{A}}_\beta = [\mathbf{a}_{\beta_1}, \ldots, \mathbf{a}_{\beta_K}]$.

Next, TFD values of the K sources at TF point (t, f) are estimated by

$$\hat{\mathcal{S}}_{\tilde{\mathbf{s}}}(t, f) \approx \tilde{\mathbf{A}}^{\#}\mathcal{S}_{\mathbf{x}}(t, f). \tag{36}$$

Here, we propose a method for estimating the mixing matrix $\mathbf{A}$. This is performed by clustering all the spatial direction vectors in (19) as for the preview TF-UBSS algorithm. Then, within each class C_i, we eliminate the far-located vectors from the centroid (in the simulation we estimate vectors $\mathbf{v}(t, f)$ such that

$$\|\mathbf{v}(t, f) - \hat{\mathbf{a}}_i\| > 0.8 \max_{\mathbf{v}(t,f) \in \Omega_i} \|\mathbf{v}(t, f) - \hat{\mathbf{a}}_i\| \tag{37}$$

leading to a size-reduced class $\tilde{C}_i$. Essentially, this is to keep the vectors corresponding to the TF region $\mathcal{R}_i$, which are ideally equal to the spatial direction $\mathbf{a}_i$ of the considered source signal. Finally, the ith column vector of $\mathbf{A}$ is estimated as the centroid of $\tilde{C}_i$.

Table IV provides a summary of the subspace projection based TF-UBSS algorithm using STFT.

V. DISCUSSION

We discuss here certain points relative to the proposed TF-UBSS algorithms and their applications.

1) Number of Sources: The number of sources N is assumed known in the clustering method (k-means) that we have used. However, there exist clustering methods [17] that perform the class estimation as well as the estimation of the number N. In our simulation, we have observed that most of the time the number of classes is overestimated, leading to poor source separation quality. Hence, robust estimation of the number of sources in the UBSS case remains a difficult open problem that deserves particular attention in future works.

2) Number of Overlapping Sources: In the subspace-based approach, we have to evaluate the number K of overlapping sources at a given TF point. This can be done by finding out the number of non-zero eigenvalues of $\mathbf{D}_{\mathbf{xx}}^{\text{wvd}}(t, f)$ using criteria such as minimum description length (MDL) or Akaike information criterion (AIC) [20]. It is also possible to consider a fixed (maximum) value of K that is used for all autosource TF points. Indeed, if the number of overlapping sources is less than K, we would estimate close-to-zero source STFT values. For example, if we assume $K = 2$ sources are present at a given TF point while only one source is effectively contributing, then we estimate one close-to-zero source STFT value. This approach increases slightly the estimation error of the source signals (especially at low SNRs) but has the advantage of simplicity compared to using information theoretic-based criterion. In our simulation, we did choose this solution with $K = 2$ or $K = 3$.

3) Quadratic Versus Linear TFDs: We have proposed two algorithms using quadratic and linear TFDs. The one using the quadratic TFD should be preferred when dealing with FM-like signals and for small or moderate sample sizes (typically up to a few hundred samples). For audio source separation often the case the sample size is large, and, hence, to reduce the computational cost, one should prefer the linear-TFD-based UBSS algorithm. Overall, the quadratic version performs slightly better than the linear one but costs much more in computations.

4) Separation Quality Versus Number of Sources: Although we are in the underdetermined case, the number of sources N should not exceed too much the number of sensors. Indeed, when N increases, the level of source interference increases, and hence, the source disjointness assumption is ill satisfied. Moreover, for a large number of sources, the likelihood of having two sources closely spaced, i.e., such that the spatial directions $\mathbf{a}_i$ and $\mathbf{a}_j$ are "close" to linear dependency, increases. In that case, vector clustering performance degrades significantly. In brief, sparseness and spatial separation are the two limiting factors against increasing the number of sources. Fig. 8

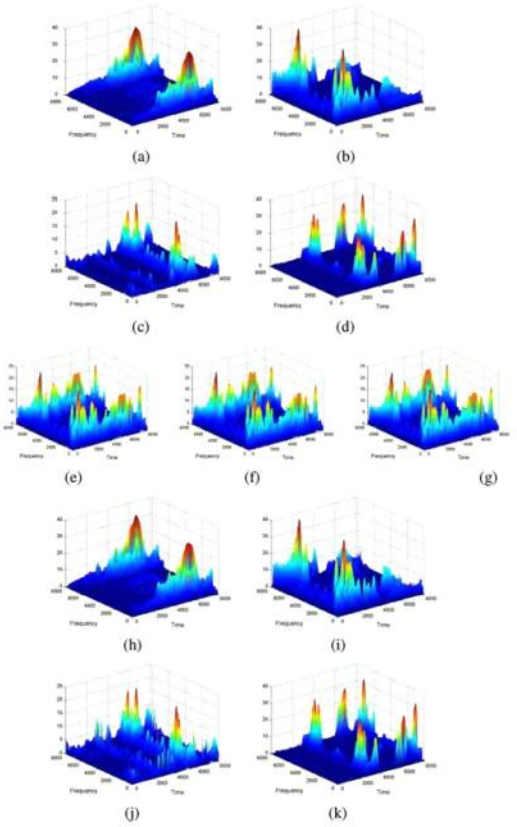


Fig. 3. Simulated example (viewed in TF domain) for the subspace-based TF-UBSS algorithm with STFT in the case of four speech sources and three sensors. The top four plots represent the original source signals, the middle three plots represent the three mixtures, and the bottom four plots represent the source estimates.

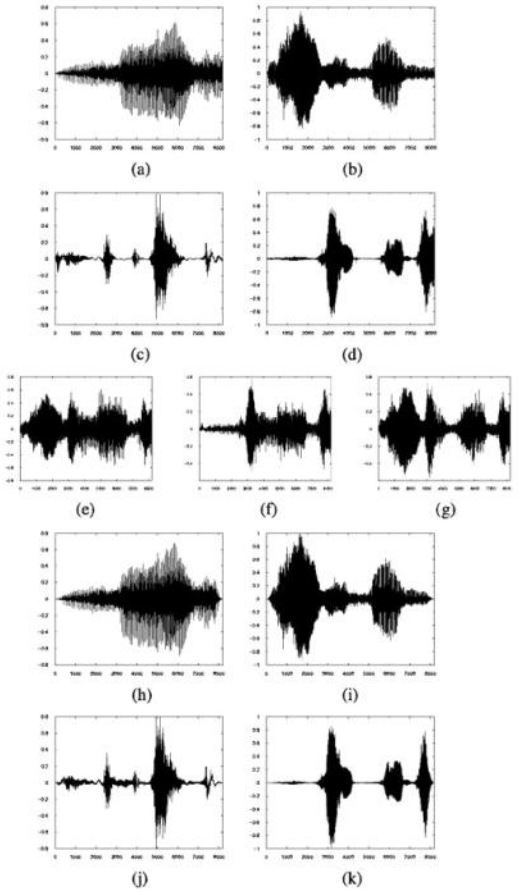


Fig. 4. Simulated example (viewed in time domain) for the subspace-based TF-UBSS algorithm with STFT in the case of four speech sources and three sensors. The top four plots (a)–(d) represent the original source signals, the middle three plots (e)–(f) represent the three mixtures, and the bottom four plots (h)–(k) represent the source estimates.

illustrates the performance degradation of source separation versus the number of sources.

VI. Simulation Results

A. Simulation Results of Subspace-Based TF-UBSS Algorithm Using STFT

In the simulations, we use a uniform linear array of $M = 3$ sensors. It receives signals from $N = 4$ independent speech sources in the far field from directions $\theta_1 = 15°, \theta_2 = 30°$ $\theta_3 = 45°$, and $\theta_4 = 75°$, respectively. The sample size is $T = 8192$ samples. In Fig. 3, the top four plots represent the TF representation of the original sources signal, the middle three plots represent the TF representation of the M mixture signals and the bottom four plots represent the TF representation of the estimate of sources by the subspace-based algorithm using STFT (Table IV). Fig. 4 represents the same disposition of signals but in the time domain.

In Fig. 5, we compare the separation performance obtained by the subspace-based algorithm with $K = 2$ and the cluster-based algorithm (Table II). It is observed that subspace-based algorithm provides much better separation results than those obtained by the cluster-based algorithm.

In the subspace-based method, one first needs to estimate the mixing matrix $\mathbf{A}$. This is done by the cluster-based method presented previously. The plot in Fig. 6 represents the normalized estimation error of $\mathbf{A}$ versus the SNR in decibels. Clearly, the proposed estimation method of the mixing matrix provides satisfactory performance, while the plot in Fig. 7 presents the separation performance when using the exact matrix $\mathbf{A}$ compared with that obtained with the proposed estimate $\hat{\mathbf{A}}$.

Fig. 8 illustrates the rapid degradation of the separation quality when we increase the number of sources from $N = 4$ to $N = 7$. This confirms the remarks made in Section V.

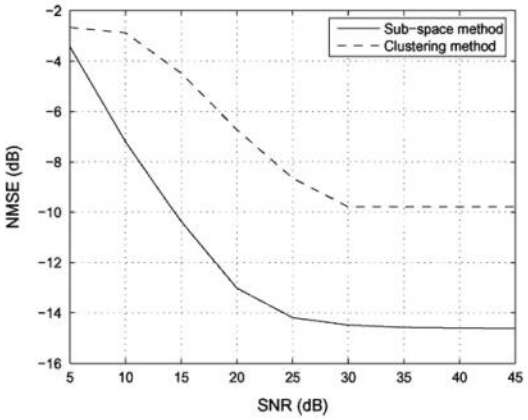


Fig. 5. Comparison between subspace-based and cluster-based TF-UBSS algorithms using STFT: normalized MSE (NMSE) versus SNR for four speech sources and three sensors.

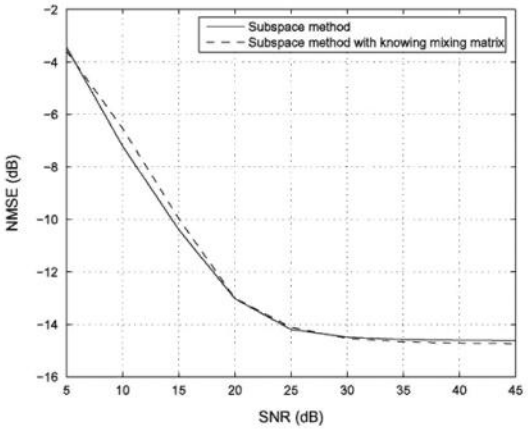


Fig. 7. Comparison, for the subspace-based TF-UBSS algorithm using STFT, when the mixing matrix $\mathbf{A}$ is known or unknown: NMSE of the source estimates.

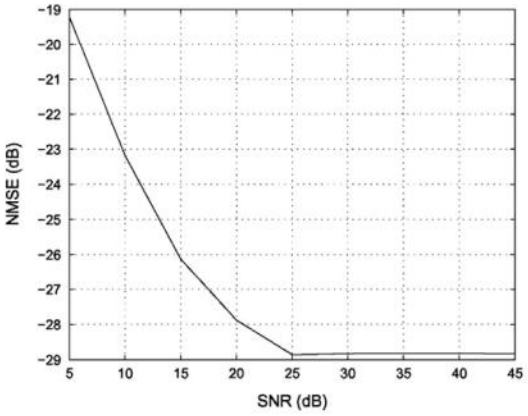


Fig. 6. Mixing matrix estimation: normalized MSE versus SNR for four speech sources and three sensors.

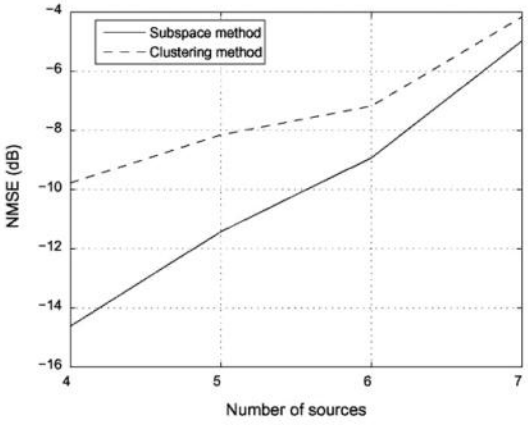


Fig. 8. Comparison between subspace-based and cluster-based TF-UBSS algorithms using STFT: NMSE versus number of sources.

In Fig. 9, we compare the performance obtained with the subspace-based method for $K = 2$ and $K = 3$. In that experiment, we have used $M = 4$ sensors and $N = 5$ source signals. One can observe that, for high SNRs, the case of $K = 3$ leads to a better separation performance than for the case of $K = 2$. However, for low SNRs, a large value of K increases the estimation noise (as mentioned in Section V) and hence degrades the separation quality.

B. Simulation Results of Subspace-Based TF-UBSS Algorithm Using STFD

In this simulation, we use a uniform linear array of $M = 3$ sensors with half wavelength spacing. It receives signals from $N = 4$ independent LFM sources, each has 256 samples, in the presence of additive Gaussian noise where the SNR = 20 dB.

We compare the cluster-based (Table I) and the proposed subspace-based (Table III) TF-UBSS algorithms. Fig. 10(a), (d), (g), and (j) represent the TFDs (using WVD) of the four sources. Fig. 10(b), (e), (h), and (k) show the estimated source TFDs using the cluster-based algorithm, whereas Fig. 10(c), (f), (i), and (l) are those obtained by the subspace-based algorithm.

From Fig. 10(b) and (e), we can see that the overlapping points between source $s_1(t)$ and source $s_2(t)$ were picked up by source $s_2(t)$ with the cluster-based algorithm. On the other hand, using the subspace-based algorithm, the intersection points have been redistributed to the two sources [Fig. 10(c) and (f)].

In general, the overlapping points in the nondisjoint case have been explicitly treated. This provides a visual performance comparison.

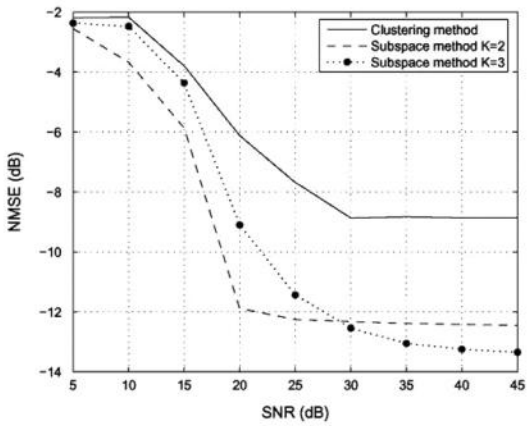


Fig. 9. Comparison between subspace-based and cluster-based TF-UBSS algorithms using STFT: NMSE of the source estimates for different sizes of the projector, for the case of five sources and four sensors.

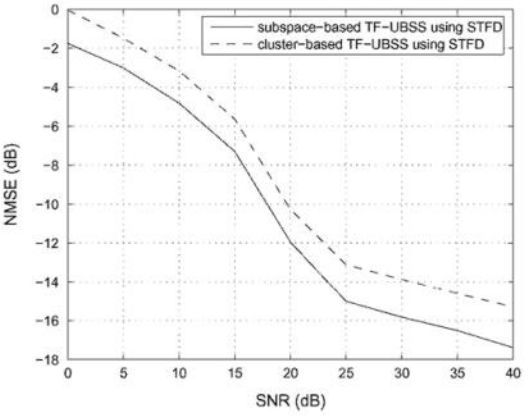


Fig. 11. Comparison between subspace-based and cluster-based TF-UBSS algorithms using STFD: normalized MSE (NMSE) versus SNR for four LFM sources and three sensors.

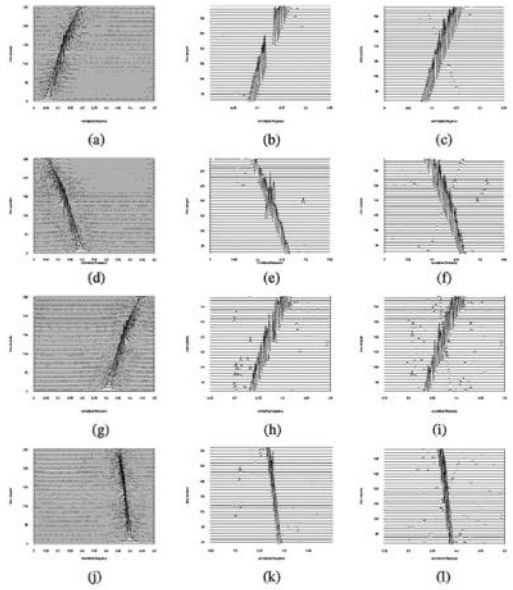


Fig. 10. Simulated example (viewed in TF domain) for the subspace-based TF-UBSS algorithm with STFT in the case of 4 LFM sources and 3 sensors. From left to right, the figures respectively represent the original source TF signatures, the estimated source TF signatures using the cluster-based algorithm, and the estimated source TF signatures using the subspace-based algorithm.

In Fig. 11, we compare the statistical separation performance between the subspace-based algorithm and the cluster-based algorithm using STFD, evaluated over 1000 Monte Carlo runs.

One can also notice that the gain here is smaller than the one obtained previously for audio sources. This is due to the fact that the overlapping region of the considered signals is smaller. This result confirms the previous visual observation with respect to the performance gain in favor of our subspace-based method.

VII. Conclusion

This paper introduces new methods for the UBSS of TF-nondisjoint nonstationary sources using time-frequency representations. The main advantages over the proposed separation algorithms are, first, a weaker assumption on the source "sparseness," i.e., the sources are not necessarily TF-disjoint, and second, an explicit treatment of the overlapping points using subspace projection, leading to significant performance improvements. Simulation results illustrate the effectiveness of our algorithms in different scenarios compared to those existing in the literature.

References

[1] A. K. Nandi, Ed., *Blind Estimation Using Higher-Order Statistics*. Boston, MA: Kluwer Academic, 1999.
[2] J.-F. Cardoso, "Blind signal separation: Statistical principles," in *Proc. IEEE*, Oct. 1998, vol. 86, no. 10, pp. 2009–2025.
[3] A. Belouchrani, K. Abed-Meraim, J.-F. Cardoso, and E. Moulines, "A blind source separation technique using second-order statistics," *IEEE Trans. Signal Process.*, vol. 45, no. 2, pp. 434–444, Feb. 1997.
[4] A. Belouchrani and M. G. Amin, "Blind source separation based on time-frequency signal representations," *IEEE Trans. Signal Process.*, vol. 46, no. 11, pp. 2888–2897, Nov. 1998.
[5] K. Abed-Meraim, Y. Xiang, J. H. Manton, and Y. Hua, "Blind source separation using second order cyclostationary statistics," *IEEE Trans. Signal Process.*, vol. 49, no. 4, pp. 694–701, Apr. 2001.
[6] P. Bofill and M. Zibulevsky, "Underdetermined blind source separation using sparse representations," *Signal Process.*, vol. 81, no. 11, pp. 2353–2362, Nov. 2001.
[7] P. O'Grady, B. Pearlmutter, and S. Rickard, "Survey of sparse and non-sparse methods in source separation," *Int. J. Imag. Syst. Tech.*, vol. 15, no. 1, pp. 18–33, 2005.
[8] N. Linh-Trung, A. Belouchrani, K. Abed-Meraim, and B. Boashash, "Separating more sources than sensors using time-frequency distributions," *EURASIP J. Appl. Signal Process.*, vol. 2005, no. 17, pp. 2828–2847, 2005.
[9] O. Yilmaz and S. Rickard, "Blind separation of speech mixtures via time-frequency masking," *IEEE Trans. Signal Process.*, vol. 52, no. 7, pp. 1830–1847, Jul. 2004.

[10] B. Barkat and K. Abed-Meraim, "Algorithms for blind components separation and extraction from the time-frequency distribution of their mixture," *EURASIP J. Appl. Signal Process.*, vol. 2004, no. 13, pp. 2025–2033, 2004.
[11] N. Linh-Trung, A. Aïssa-El-Bey, K. Abed-Meraim, and A. Belouchrani, "Underdetermined blind source separation of non-disjoint nonstationary sources in time-frequency domain," in *Proc. Int. Symp. Signal Processing Its Applications (ISSPA)*, Sydney, Australia, Aug. 2005, vol. 1, pp. 46–49.
[12] S. Rickard, T. Melia, and C. Fearon, "Desprit—Histogram based blind source separation of more sources than sensors using subspace methods," in *Proc. IEEE Workshop on Applications Signal Processing Audio Acoustics*, Oct. 2005, pp. 5–8.
[13] B. Boashash, Ed., *Time Frequency Signal Analysis and Processing: Method and Applications*. Oxford, U.K.: Elsevier, 2003.
[14] B. Barkat and B. Boashash, "A high-resolution quadratic time-frequency distribution for multicomponent signal analysis," *IEEE Trans. Signal Process.*, vol. 49, no. 10, pp. 2232–2239, Oct. 2001.
[15] L. D. Lathauwer, B. Moor, and J. Vandewalle, "ICA techniques for more sources than sensors," in *Proc. IEEE Signal Processing Workshop on Higher Order Statistics*, Jun. 1999, pp. 121–124.
[16] G. F. Boudreaux-Bartels and T. W. Parks, "Time-varying filtering and signal estimation using Wigner distributions," *IEEE Trans. Acoust., Speech, Signal Process.*, vol. ASSP-34, no. 3, pp. 442–451, Mar. 1986.
[17] I. E. Frank and R. Todeschini, *The Data Analysis Handbook*. New York: Elsevier, Sci., 1994.
[18] D. W. Griffin and J. S. Lim, "Signal estimation from modified short-time Fourier transform," *IEEE Trans. Acoustic, Speech, Signal Process.*, vol. ASSP-32, no. 2, pp. 236–243, Apr. 1984.
[19] M. Zibulevsky, B. A. Pearlmutter, P. Bofill, and P. Kisilev, *Independent Component Analysis: Principles and Practice*, S. J. Roberts and R. M. Everson, Eds. Cambridge, U.K.: Cambridge Univ. Press, 2001, ch. Blind Source Separation by Sparse Decomposition.
[20] M. Wax and T. Kailath, "Detection of signals by information theoretic criteria," *IEEE Trans. Acoust., Speech, Signal Process.*, vol. ASSP-33, no. 2, pp. 387–392, Apr. 1985.

Abdeldjalil Aïssa-El-Bey was born in Algiers, Algeria, in 1981. He received the State Engineering degree from École Nationale Polytechnique (ENP), Algiers, Algeria, in 2003 and the M.S. degree in signal processing from Supélec and Paris XI University, Orsay, France, in 2004. Currently he is working towards the Ph.D. degree at the Signal and Image Processing Department of École Nationale Supérieure des Télécommunications (ENST) Paris, France.

His research interests are blind source separation, blind system identification and equalization, statistical signal processing, wireless communications, and adaptive filtering.

Nguyen Linh-Trung was born in Vietnam in 1973. He received the B.E.E. degree and Ph.D. degree in electrical engineering from the Queensland University of Technology, Brisbane, Australia, in 1997 and 2002, respectively

He has visited the École Nationale Supérieure des Télécommunications, Paris, France, several times (in 2001, 2002, and 2003) during and after his Ph.D., where he worked on the problem of time-frequency based underdetermined blind source separation. From October 2002 to January 2003, he was a Postdoctoral Research Associate with the Information Group of the Aston University, Birmingham, U.K., where he worked on optimal biorthogonal representation of signals. From September 2003 to September 2005, he was a Postdoctoral Research Fellow with the Centre National d'Études Spatiales, Toulouse, France, where he investigated mechanisms for priority access in emergency communications over public satellite networks. Since January 2006, he has been a faculty member at the College of Technology of the Vietnam National University, Hanoi.

Karim Abed-Meraim (SM'04) was born in 1967. He received the State Engineering degree from École Polytechnique, Paris, France, in 1990, the State Engineering degree from École Nationale Supérieure des Télécommunications (ENST) Paris, France, in 1992, the M.S. degree from Paris XI University, Orsay, France, in 1992, and the Ph.D. degree from ENST in 1995.

From 1995 to 1998, he was a Research Staff Member with the Electrical Engineering Department of the University of Melbourne, Melbourne, Australia, where he worked on several research projects related to blind system identification for wireless communications, blind source separation, and array processing for communications. Since 1998, he has been an Associate Professor with the Signal and Image Processing Department of ENST. His research interests are in signal processing for communications and include system identification, multiuser detection, space–time coding, adaptive filtering and tracking, array processing, and performance analysis.

Adel Belouchrani received the State Engineering degree from École Nationale Polytechnique (ENP), Algiers, Algeria, in 1991, the M.S. degree in signal processing from the Institut National Polytechnique de Grenoble (INPG), Grenoble, France, in 1992, and the Ph.D. degree in signal and image processing from Télécom (ENST) Paris, France, in 1995.

He was a Visiting Scholar at the Electrical Engineering and Computer Sciences Department, University of California, Berkeley, from 1995 to 1996. He was with the Department of Electrical and Computer Engineering, Villanova University, Villanova, PA, as a Research Associate from 1996 to 1997. He also served as a Consultant to Comcast, Inc., Philadelphia, PA, during the same period. From August 1997 to October 1997, he was with Alcatel ETCA, Belgium. Since 1998, he has been with the Electrical Engineering Department of ENP first as an Associate Professor, and then Professor since 2006. His research interests are in statistical signal processing and (blind) array signal processing with applications in biomedical and communications, time-frequency analysis, time-frequency array signal processing, and wireless and spread spectrum communications.

Yves Grenier (M'81) was born in Ham, Somme, France, in 1950. He received the Ingénieur degree from École Centrale de Paris, Paris, France, in 1972, the Docteur-Ingénieur degree from École Nationale Supérieure des Télécommunications, Paris, France, in 1977, and the Doctorat d'État es Sciences Physiques from the University of Paris-Sud, Paris, France, in 1984.

Since 1977, he has been with École Nationale Supérieure des Télécommunications, Paris, France, first as an Assistant Professor and then as a Professor since 1984. He has been Head of the Signal and Image Processing Department since January 2005. Until 1979, his interests were in speech recognition, speaker identification, and speaker adaptation of recognition systems. He then began working on signal modeling, spectral analysis of noisy signals, with applications in speech recognition and synthesis, estimation of nonstationary models, and time-frequency representations. He is presently interested in audio signal processing (acoustic echo cancellation, noise reduction, signal separation, microphone arrays, and loudspeaker arrays).

Dr. Grenier is a member of the Audio Engineering Society (AES).

About this section

Examples of what to put in this section
Use this section for any notes that don't fit well into any of the other sections in your notebook. If you find yourself putting many related notes in this section, create a new section related to that topic.

Tips

- To quickly find what you are looking for in OneNote 2007, use the Find feature (**CTRL+F**) . OneNote 2007 searches text, text within images, ink writing, and audio recordings.
- Want to start organizing some of these pages? Click the page tabs that you want to move, and then drag them to the notebook or section where you want to put them. If you change your mind, you can drag them back to this section.