REAL-TIME BLIND EXTRACTION OF DOMINANT TARGET SOURCES FROM MANY BACKGROUND INTERFERENCE SOURCES

Hiroshi Sawada *Ryo Mukai* *Shoko Araki* *Shoji Makino*

NTT Communication Science Laboratories, NTT Corporation
2-4 Hikaridai, Seika-cho, Soraku-gun, Kyoto 619-0237, Japan
{sawada,ryo,shoko,maki}@cslab.kecl.ntt.co.jp

ABSTRACT

This paper presents a method for enhancing target sources of interest and suppressing background interference sources. The target sources are assumed to be close to sensors, to have dominant powers at the sensors, and to have non-Gaussianity. The method is based on a two-stage process where independent component analysis (ICA) is first employed in each frequency bin and then time-frequency masking is used to improve the performance further. We propose a new sophisticated method for deciding the number of target sources and then selecting their frequency components. Experimental results for cocktail party situations are presented to show the effectiveness of the method. We also describe a real-time implementation of the method.

1. INTRODUCTION

The technique for estimating individual source components from their mixtures at sensors is known as blind source separation (BSS) [1, 2]. With audio applications such as speech enhancement, the sources do not necessarily have equal significance. We often want to extract only a few sources that are close to sensors, have dominant powers, and/or have interesting features.

Based on this idea, we have proposed a method for blindly extracting one dominant target source and suppressing the interference sources [3]. This paper generalizes the method [3] so that the number of target sources can be more than one. Let us formulate the task. Suppose that a few close target sources $\mathrm{s}_1, \ldots, \mathrm{s}_Q$ and other background sources $\mathrm{s}_{Q+1}, \ldots, \mathrm{s}_N$ are convolutively mixed and observed at $M \geq 2$ sensors

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

where $\mathrm{h}_{jk}(l)$ represents the impulse response from source k to sensor j. The goal is to have output signals $\mathrm{y}_i(t)$, $i = 1, \ldots, Q$ that are the components of s_i measured at a selected sensor J:

$$\mathrm{x}_{Ji}(t) = \sum_{l} \mathrm{h}_{Ji}(l)\, \mathrm{s}_i(t-l), \quad i = 1, \ldots, Q. \quad (2)$$

Note that $\mathrm{x}_j(t) = \sum_{k=1}^{N} \mathrm{x}_{jk}(t)$. The task should be performed only with the M observed signals $\mathrm{x}_1, \ldots, \mathrm{x}_M$. The number of target sources Q and the total number of sources N are unknown. The number Q is assumed to be no more than the number of sensors M, and N may be larger than M.

The first issue is how to extract target sources $\mathrm{s}_1, \ldots, \mathrm{s}_Q$ blindly. Even if the total number of sources N could be larger than M, independent component analysis (ICA) [1, 2] with an $N = M$ assumption produces M components that maximize an ICA criterion. We assume that the target sources are non-Gaussian, close to sensors, and dominant in the mixtures. Therefore, we expect that some of the M components correspond to target sources $\mathrm{s}_1, \ldots, \mathrm{s}_Q$ whose ICA criteria are high.

We employ ICA in the frequency domain [4–7]. This is because it is efficient [7] and can also be used with time-frequency masking, which is discussed below. An additional operation that should be performed is the selection of each target component in every frequency bin. This is known as the permutation problem of frequency-domain BSS [8]. We solve this problem by clustering the basis vectors (6) produced by ICA after some normalization. Moreover, the number Q of dominant target sources can be decided by examining the variances of the clusters.

The next issue is that some interference still remains in the extracted frequency components when $N > M$. Time-frequency masking is known to be efficient for sources with sparseness in the time-frequency domain, such as speech. The performance depends on how well we can specify the time-frequency slots where the target source is active. Although many methods [9–12] for specifying masks have been proposed, they are basically applied to two sensors. We propose a new criterion for specifying masks that can be applied to any number (≥ 2) of sensors.

This paper also presents a real-time implementation of the proposed method. More accurately, it is a blockwise batch implementation [13], where ICA is applied to a block of a few seconds but the system input-output delay is kept at less than a second.

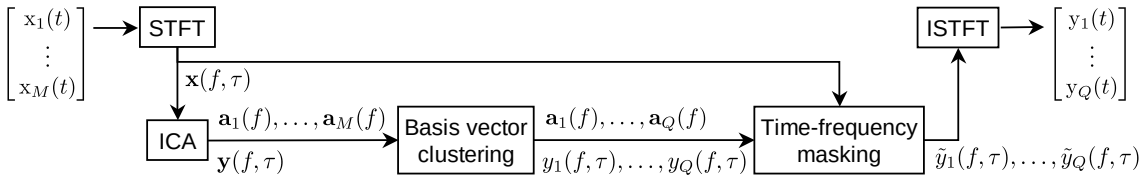

Figure 1: *Flow of proposed method*

2. FLOW OF PROPOSED METHOD

Figure 1 shows the flow of the proposed method. This section explains each part except "Basis vector clustering", which is explained in more detail in Sec. 3.

2.1. Frequency domain operations

First, time-domain observed signals $\mathrm{x}_j(t)$ sampled at frequency f_s are converted into frequency-domain time-series signals $x_j(f,\tau)$ with an L-point STFT:

$$x_j(f,\tau) = \sum_{r=-L/2}^{L/2-1} \mathrm{x}_j(\tau+r)\,\mathrm{win}(r)\,e^{-\jmath 2\pi fr}, \qquad (3)$$

where $f \in \{0,\ \frac{1}{L}f_s,\ \ldots,\ \frac{L-1}{L}f_s\}$ is a frequency, $\mathrm{win}(r)$ is a window that tapers smoothly to zero at each end, and τ is a new index representing time.

The following operations are performed in the frequency domain. There are two advantages to this. First, the convolutive mixtures (1) can be approximated as instantaneous mixtures at each frequency:

$$x_j(f,\tau) \approx \sum_{k=1}^{N} h_{jk}(f)s_k(f,\tau), \qquad (4)$$

where $h_{jk}(f)$ is the frequency response from source k to sensor j, and $s_k(f,\tau)$ is a frequency-domain time-series signal of $\mathrm{s}_k(t)$ obtained by the same operation as (3). The second advantage is that the sparseness of a source signal becomes prominent in the time-frequency domain if the source is colored and non-stationary such as speech.

After several operations, which are explained in the following subsections and in Sec. 3, time-domain output signals $\mathrm{y}_i(t)$ are obtained by an inverse STFT (ISTFT):

$$\mathrm{y}_i(\tau+r) = \frac{1}{L\cdot\mathrm{win}(r)} \sum_{f\in\{0,\ \frac{1}{L}f_s,\ \ldots,\ \frac{L-1}{L}f_s\}} \tilde{y}_i(f,\tau)\,e^{\jmath 2\pi fr}$$

for $i = 1,\ldots,Q$.

2.2. Independent component analysis (ICA)

To extract the components of dominant sources, we apply ICA to observation vectors $\mathbf{x}(f,\tau) = [x_1(f,\tau),\ldots, x_M(f,\tau)]^T$ by assuming that the number of independent components is equal to M:

$$\mathbf{y}(f,\tau) = \mathbf{W}(f)\,\mathbf{x}(f,\tau), \qquad (5)$$

where $\mathbf{W}$ is an $M\times M$ separation matrix and $\mathbf{y}(f,\tau) = [y_1(f,\tau),\ldots,y_M(f,\tau)]^T$ is the vector of independent components. Then, we calculate the inverse of $\mathbf{W}$ to obtain basis vectors

$$[\mathbf{a}_1,\cdots,\mathbf{a}_M] = \mathbf{W}^{-1},\ \mathbf{a}_i = [a_{1i},\ldots,a_{Mi}]^T. \qquad (6)$$

By multiplying both sides of (5) by $\mathbf{W}^{-1}$, the observation vector $\mathbf{x}(f,\tau)$ is represented by a linear combination of basis vectors $\mathbf{a}_1,\ldots,\mathbf{a}_M$:

$$\mathbf{x}(f,\tau) = \sum_{i=1}^{M} \mathbf{a}_i(f)y_i(f,\tau). \qquad (7)$$

Some of these independent components correspond to the components of dominant sources. However, the correspondence is not clear at this stage because of the permutation ambiguity of ICA. Thus, basis vector clustering (Sec. 3) is performed to solve this permutation ambiguity.

2.3. Time-frequency masking

In a general case where the total number of sources N is larger than the number of sensors M, independent components $y_1(f,\tau),\ldots,y_Q(f,\tau)$ produced by ICA have some residuals caused by the limitation of spatial filtering [3]. Time-frequency masking is used to reduce such residuals, and is performed by

$$\tilde{y}_i(f,\tau) = \mathcal{M}_i(f,\tau)\,y_i(f,\tau),\ \ i = 1,\ldots,Q \qquad (8)$$

where $0 \le \mathcal{M}_i(f,\tau) \le 1$ is a mask specified for each time-frequency slot (f,τ). The masks should have large values for time-frequency slots where the target source is active. We specify masks based on the angle $\theta_i(f,\tau)$ between $\mathbf{a}_i(f)$ and $\mathbf{x}(f,\tau)$ calculated in the space transformed by a whitening matrix $\mathbf{V}(f) = \mathbf{R}^{-1/2}$, $\mathbf{R} = \langle\mathbf{x}(\tau)\,\mathbf{x}(\tau)^H\rangle_\tau$. The angle is calculated by

$$\theta_i(f,\tau) = \arccos\frac{|\mathbf{b}_i^H(f)\,\mathbf{z}(f,\tau)|}{||\mathbf{b}_i(f)||\cdot||\mathbf{z}(f,\tau)||}, \qquad (9)$$

where $\mathbf{z}(f,\tau) = \mathbf{V}(f)\mathbf{x}(f,\tau)$ is a whitened sample and $\mathbf{b}_i(f) = \mathbf{V}(f)\mathbf{a}_i(f)$ is the basis vector in the whitened space. The angle would be close to 0 degrees if the target source is active in the time-frequency slot. Then, we

calculate a mask by using a logistic function

$$\mathcal{M}_i(\theta_i(f,\tau)) = \frac{1}{1+e^{g(\theta_i-\theta_T)}}, \qquad (10)$$

where θ_T and g are parameters specifying the transition point and its steepness, respectively. Typical values are $(\theta_T, g) = (0.333\pi, 20)$. As θ_T becomes smaller, the residual power that appears in $\tilde{y}_i$ decreases but the musical noise in y_i increases.

3. BASIS VECTOR CLUSTERING

Basis vector clustering has two purposes: to solve the permutation ambiguity in (7) and to select the target sources. The previous method [3] selects the cluster with the minimum variance as the only target source. This paper generalizes this idea so that the system decides the number Q of dominant target sources, and then selects those target sources.

We normalize all basis vectors $\mathbf{a}_i(f)$, $i = 1, \ldots, M$, for all frequency bins $f = 0, \frac{1}{L}f_s, \ldots, \frac{L-1}{L}f_s$ such that they form clusters, each of which corresponds to an individual source. This normalization is performed by selecting a reference sensor J and calculating

$$\bar{a}_{ji}(f) \leftarrow |a_{ji}(f)| \exp\left[\jmath \frac{\arg[a_{ji}(f)/a_{Ji}(f)]}{4fc^{-1}d_{\max}}\right] \qquad (11)$$

where c is the propagation velocity and $d_{\max}$ is the maximum distance between the sensor J and a sensor $^{\forall}j \in \{1, \ldots, M\}$. We then apply unit-norm normalization

$$\bar{\mathbf{a}}_i(f) \leftarrow \bar{\mathbf{a}}_i(f) \,/\, ||\bar{\mathbf{a}}_i(f)|| \qquad (12)$$

to $\bar{\mathbf{a}}_i(f) = [\bar{a}_{1i}(f), \ldots, \bar{a}_{Mi}(f)]^T$.

After normalizing all the basis vectors, we employ a clustering algorithm to find clusters $C_1, \ldots, C_M$ formed by normalized vectors $\bar{\mathbf{a}}_i(f)$. The centroid $\mathbf{c}_k$ of a cluster C_k is calculated by

$$\mathbf{c}_k \leftarrow \sum_{\bar{\mathbf{a}} \in C_k} \bar{\mathbf{a}}/|C_k|, \quad \mathbf{c}_k \leftarrow \mathbf{c}_k/||\mathbf{c}_k||, \qquad (13)$$

where $|C_k|$ is the number of vectors in C_k. The clustering criterion is to minimize the total sum $\mathcal{J}$ of the squared distances between cluster members and their centroid

$$\mathcal{J} = \sum_{k=1}^{M} \mathcal{J}_k, \;\; \mathcal{J}_k = \sum_{\bar{\mathbf{a}} \in C_k} ||\bar{\mathbf{a}} - \mathbf{c}_k||^2. \qquad (14)$$

This minimization can be performed efficiently with the k-means clustering algorithm [14].

Once we have found M clusters $C_1, \ldots, C_M$, we need to identify the clusters that correspond to dominant target sources $s_1, \ldots, s_Q$. We decide that a cluster C_k with a small variance $\mathcal{J}_k/|C_k|$ belongs to the set of target sources. The rationale behind this criterion is that the direct-path (nearfield) mixing model is more valid for $s_1, \ldots, s_Q$ than for the other sources [3].

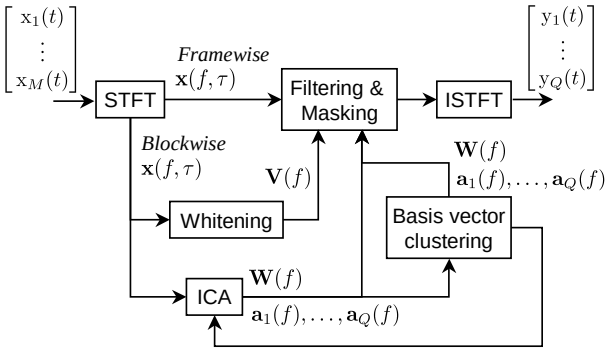

Figure 2: *System structure for a real-time implementation*

To identify target source clusters, we sort $C_1, \ldots, C_M$ so that their variances are arranged in ascending order:

$$\frac{\mathcal{J}_1}{|C_1|} \leq \cdots \leq \frac{\mathcal{J}_Q}{|C_Q|} \leq th_{var} \leq \cdots \leq \frac{\mathcal{J}_M}{|C_M|}, \qquad (15)$$

where th_{var} is a predefined threshold for specifying the set of target sources. Then, to align the permutation ambiguities, we renumber the indexes of the basis vectors by

$$\mathbf{a}_k(f) \leftarrow \mathbf{a}_{\Pi_f(k)}(f), \qquad (16)$$

where $\Pi_f : \{1, \ldots, Q\} \rightarrow \{1, \ldots, M\}$ is a one-to-one mapping decided for each frequency f by

$$\Pi_f = \mathrm{argmin}_{\Pi} \sum_{k=1}^{Q} ||\bar{\mathbf{a}}_{\Pi(k)}(f) - \mathbf{c}_k||^2. \qquad (17)$$

We also renumber independent components $y_1(f,\tau), \ldots, y_M(f,\tau)$ accordingly.

4. REAL-TIME IMPLEMENTATION

This section describes a real-time implementation of the method. Our current implementation is mainly with Matlab and partially with C, and performed on an ordinary desktop or notebook computer that has a single CPU. Thus, we employ a blockwise batch algorithm [13]. Figure 2 shows the system structure. The upper path, which consists of STFT, Filtering & Masking and ISTFT, is operated framewise, where the input-output delay is kept as small as a few STFT frames. The lower path, which includes Whitening, ICA, and Basis vector clustering, is operated blockwise to ensure learning convergence and also to interleave operations for many frequency bins. A typical configuration is as follows: STFT frame size = 128 ms, frame shift = 32 ms, learning block size = 98 frames (3.232 s), block interval = 49 frames (1.568 s). This configuration means that Whitening and ICA for each frequency bin and basis vector clustering are performed every 1.568 s.

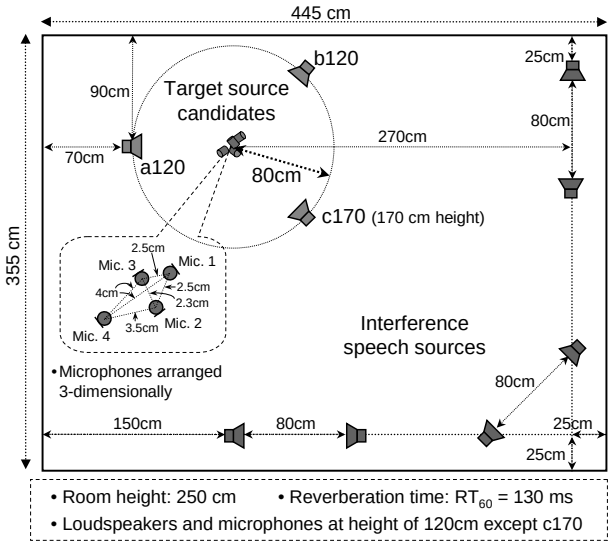

Figure 3: *Experimental conditions*

5. EXPERIMENT

To evaluate the separation performance for multi-target cases, we performed experiments in a batch mode. We measured impulse responses $\mathrm{h}_{jk}(l)$ under the conditions shown in Fig. 3. The speaker positions simulated a cocktail party situation. Mixtures at the microphones were made by convolving the impulse responses and 6-second English and Japanese speech sampled at 8 kHz. We used four 3-dimensionally arranged microphones ($M=4$). The system knew only the maximum distance (4 cm) between the reference microphone (Mic. 1) and the others. Experiments were conducted with 10 combinations of 9 speech (3 targets + 6 background interference sources). We used $th_{var}=0.015$ to specify the set of target sources by (15).

Table 1 shows the average signal-to-interference ratio (SIR) improvements obtained solely with ICA, and with a combination of ICA and T-F masking. Even in such hard situations, the system succeeded in enhancing and separating the target sources.

6. CONCLUSION

This paper described some extensions of our previously reported method [3] that blindly extracts a close target source from many background interference sources. The first extension makes it possible for the system to handle plural target sources by examining the cluster variances of normalized basis vectors. The second extension involves the realization of a real-time implementation.

Table 1: Average SIR improvement (dB)

Target position	a120	b120	c170
$\mathsf{InputSIR}_i$	-3.9	-3.6	-5.9
Only ICA	12.5	13.6	14.5
ICA and T-F masking	15.1	16.5	17.6

7. REFERENCES

[1] S. Haykin, Ed., *Unsupervised Adaptive Filtering (Volume I: Blind Source Separation)*, John Wiley & Sons, 2000.

[2] A. Hyvärinen, J. Karhunen, and E. Oja, *Independent Component Analysis*, John Wiley & Sons, 2001.

[3] H. Sawada, S. Araki, R. Mukai, and S. Makino, "Blind extraction of a dominant source signal from mixtures of many sources," in *Proc. ICASSP 2005*, Mar. 2005, vol. III, pp. 61–64.

[4] P. Smaragdis, "Blind separation of convolved mixtures in the frequency domain," *Neurocomputing*, vol. 22, pp. 21–34, 1998.

[5] L. Parra and C. Spence, "Convolutive blind separation of non-stationary sources," *IEEE Trans. Speech Audio Processing*, vol. 8, no. 3, pp. 320–327, May 2000.

[6] L. Schobben and W. Sommen, "A frequency domain blind signal separation method based on decorrelation," *IEEE Trans. Signal Processing*, vol. 50, no. 8, pp. 1855–1865, Aug. 2002.

[7] H. Sawada, R. Mukai, S. Araki, and S. Makino, "Frequency-domain blind source separation," in *Speech Enhancement*, J. Benesty, S. Makino, and J. Chen, Eds., pp. 299–327. Springer, Mar. 2005.

[8] H. Sawada, R. Mukai, S. Araki, and S. Makino, "A robust and precise method for solving the permutation problem of frequency-domain blind source separation," *IEEE Trans. Speech Audio Processing*, vol. 12, no. 5, pp. 530–538, Sept. 2004.

[9] Ö. Yilmaz and S. Rickard, "Blind separation of speech mixtures via time-frequency masking," *IEEE Trans. Signal Processing*, vol. 52, no. 7, pp. 1830–1847, July 2004.

[10] S. Araki, S. Makino, A. Blin, R. Mukai, and H. Sawada, "Underdetermined blind separation for speech in real environments with sparseness and ICA," in *Proc. ICASSP 2004*, May 2004, vol. III, pp. 881–884.

[11] N. Roman and D. Wang, "Binaural sound segregation for multisource reverberant environments," in *Proc. ICASSP 2004*, May 2004, vol. II, pp. 373–376.

[12] D. Kolossa and R. Orglmeister, "Nonlinear postprocessing for blind speech separation," in *Proc. ICA 2004 (LNCS 3195)*, Sept. 2004, pp. 832–839.

[13] R. Mukai, H. Sawada, S. Araki, and S. Makino, "Blind source separation for moving speech signals using blockwise ICA and residual crosstalk subtraction," *IEICE Trans. Fundamentals*, vol. E87-A, no. 8, pp. 1941–1948, Aug. 2004.

[14] R. O. Duda, P. E. Hart, and D. G. Stork, *Pattern Classification*, Wiley Interscience, 2nd edition, 2000.