A general framework for online audio source separation

Laurent S. R. Simon, Emmanuel Vincent

A General Framework for Online Audio Source Separation

Laurent S. R. Simon and Emmanuel Vincent

INRIA, Centre de Rennes - Bretagne Atlantique
Campus de Beaulieu, 35042 Rennes Cedex, France
{laurent.s.simon@inria.fr, emmanuel.vincent@inria.fr}

Abstract. We consider the problem of online audio source separation. Existing algorithms adopt either a sliding block approach or a stochastic gradient approach, which is faster but less accurate. Also, they rely either on spatial cues or on spectral cues and cannot separate certain mixtures. In this paper, we design a general online audio source separation framework that combines both approaches and both types of cues. The model parameters are estimated in the Maximum Likelihood (ML) sense using a Generalised Expectation Maximisation (GEM) algorithm with multiplicative updates. The separation performance is evaluated as a function of the block size and the step size and compared to that of an offline algorithm.

Keywords: Online audio source separation, nonnegative matrix factorisation, sliding block, stochastic gradient.

1 Introduction

Audio source separation is the process of recovering a set of audio signals from a given mixture signal. This can be addressed via established approaches such as Independent Component Analysis (ICA), binary masking and Sparse Component Analysis (SCA) [1] or more recent approaches such as local Gaussian modeling and Nonnegative Matrix Factorisation (NMF) [2]. Most current algorithms are offline algorithms which require the whole signal in order to estimate the sources. In this paper, we focus on online audio source separation, whereby only the past samples of the mixture are available. This constraint arises in particular in real-time scenarios.

A few online implementations have been designed for ICA [3] [4], time-frequency masking [5], local Gaussian modeling [6], spectral continuity-based separation [7] and NMF [8]. However, these algorithms rely either on spatial cues [3] – [6] or on spectral cues [7, 8] alone. Such algorithms are not capable of separating mixtures where several sources have the same spatial position and several sources have similar spectral characteristics. For example, in pop music, the voice, the snare drum, the bass drum and the bass are often mixed to the centre and several voices or several guitars are present.

In order to address this issue, we consider the general flexible source separation framework in [9]. This framework generalises a wide range of algorithms such as certain forms of ICA, local Gaussian modeling and NMF, and enables the specification of additional constraints on the sources such as harmonicity. By jointly exploiting spatial and spectral cues, it makes it possible to robustly separate difficult mixtures such as above.

The two main approaches for online source separation are the sliding block (also known as blockwise) approach, as used in [3] [4] [5] [7], and the stochastic gradient (also known as stepwise) approach, as used in [6] [8]. The sliding block method consists in applying the offline audio source separation algorithm to a block of M time frames. Once this block of signal has been processed, a frame is extracted for each of the J sources before sliding the processing block by one frame. This approach is computationally intensive but accurate. The stepwise method offers to update the model parameters in every frame using only the latest available frame and the model parameters estimated in the previous frame. As it uses only the latest available frame at a given time, this approach is faster than the sliding block approach but can be inaccurate.

In this paper, we propose a general iterative online algorithm for the source separation framework in [9] that combines the sliding block approach and the stepwise approach using two hyper-parameters: the block size M and the step size α. As a by-product, we provide a way of circumventing the annealing procedure in [9], which would require a large number of iterations per block. Moreover, we determine the best trade-off between these two approaches experimentally on a set of real-world music mixtures.

The structure of the rest of the paper is as follows: the flexible framework in [9] is introduced in Section 2. Section 3 presents the online algorithm. Experimental results are shown in Section 4. The conclusion can be found in Section 5.

2 General audio source separation framework

We operate in the time-frequency (TF) domain by means of the Short-Time Fourier Transform (STFT). In each frequency bin f and each time frame n, the multichannel mixture signal $\mathbf{x}(f,n)$ can be expressed as

$$\mathbf{x}(f,n) = \sum_{j=1}^{J} \mathbf{c}_j(f,n) \tag{1}$$

where J is the number of sources and $\mathbf{c}_j(f,n)$ is the STFT of the spatial image of the j-th source.

2.1 Model

We assume that $\mathbf{c}_j(f,n)$ is a complex-valued Gaussian random vector with zero mean and covariance matrix $\mathbf{R}_{\mathbf{c}_j}(f,n)$

$$\mathbf{c}_j \sim \mathcal{N}_c(\mathbf{0}, \mathbf{R}_{\mathbf{c}_j}) \tag{2}$$

and that $\mathbf{R}_{\mathbf{c}_j}(f,n)$ factors as

$$\mathbf{R}_{\mathbf{c}_j}(f,n) = \mathbf{R}_j(f) v_j(f,n) \tag{3}$$

where $\mathbf{R}_j(f)$ is the spatial covariance matrix of the j-th source and $v_j(f,n)$ is its spectral variance.

In [9], $\mathbf{R}_j(f)$ is expressed as $\mathbf{R}_j(f) = \mathbf{A}_j(f)\mathbf{A}_j^H(f)$, and $\mathbf{A}_j(f)$ is estimated instead. This results in an annealing procedure, which would translate into a large number of iterations within each block in our context. In order to circumvent the annealing, we assume that $\mathbf{R}_j(f)$ is full-rank and directly estimate $\mathbf{R}_j(f)$ instead, similarly to [10].

The spectral variance $v_j(f,n)$ is modeled via a form of hierarchical NMF [9]. The matrix of spectral variances $\mathbf{V}_j \triangleq [v_j(f,n)]_{f,n}$ is first decomposed into the product of an excitation spectral power $\mathbf{V}_j^{\mathrm{x}}$ and a filter spectral power $\mathbf{V}_j^{\mathrm{f}}$

$$\mathbf{V}_j = \mathbf{V}_j^{\mathrm{x}} \odot \mathbf{V}_j^{\mathrm{f}} \tag{4}$$

where $\odot$ denotes entrywise multiplication. $\mathbf{V}_j^{\mathrm{x}}$ is further decomposed into the product of a matrix of narrowband spectral patterns $\mathbf{W}_j^{\mathrm{x}}$, a matrix of spectral envelope weights $\mathbf{U}_j^{\mathrm{x}}$, a matrix of temporal envelope weights $\mathbf{G}_j^{\mathrm{x}}$ and a matrix of time-localised temporal patterns $\mathbf{H}_j^{\mathrm{x}}$, so that

$$\mathbf{V}_j^{\mathrm{x}} = \mathbf{W}_j^{\mathrm{x}} \mathbf{U}_j^{\mathrm{x}} \mathbf{G}_j^{\mathrm{x}} \mathbf{H}_j^{\mathrm{x}}. \tag{5}$$

$\mathbf{V}_j^{\mathrm{f}}$ is decomposed in a similar way.

This factorisation enables the specification of various spectral or temporal constraints over the sources. For example, harmonicity can be enforced by fixing $\mathbf{W}_j^{\mathrm{x}}$ to a set of narrowband harmonic patterns.

2.2 Offline EM-MU algorithm

In an offline context, the model parameters are estimated in the Maximum Likelihood (ML) sense by a Generalised Expectation-Maximisation (GEM) algorithm combined with Multiplicative Updates (MU) applied to the complete data $\{\mathbf{c}_j(f,n)\}$.

The log-likelihood is defined using the empirical mixture covariance matrix $\widehat{\mathbf{R}}_{\mathbf{x}}(f,n)$ [10] as

$$\log \mathcal{L} = \sum_{f,n} -\operatorname{tr}\left(\mathbf{R}_{\mathbf{x}}^{-1}(f,n)\widehat{\mathbf{R}}_{\mathbf{x}}(f,n)\right) - \log\det(\pi \mathbf{R}_{\mathbf{x}}(f,n)) \tag{6}$$

where

$$\mathbf{R}_{\mathbf{x}}(f,n) = \sum_{j=1}^{J} \mathbf{R}_{\mathbf{c}_j}(f,n) \tag{7}$$

is the covariance of the mixture $\mathbf{x}(f,n)$.

In the E-step, the expectation of the natural statistics is computed via [10]

$$\mathbf{\Omega}_j(f,n) = \mathbf{R}_{\mathbf{c}_j}(f,n)\mathbf{R}_{\mathbf{x}}^{-\mathbf{1}}(f,n) \tag{8}$$

$$\widehat{\mathbf{R}}_{\mathbf{c}_j}(f,n) = \mathbf{\Omega}_j(f,n)\widehat{\mathbf{R}}_{\mathbf{x}}(f,n)\mathbf{\Omega}_j^H(f,n) + (\mathbf{I} - \mathbf{W}_j(f,n))\mathbf{R}_{\mathbf{c}_j}(f,n) \tag{9}$$

where $\mathbf{\Omega}_j$ is the Wiener filter, $\mathbf{I}$ is the $I \times I$ identity matrix and I is the number of channels of the mixture.

In the M-step, the model parameters are updated as [9, 10]

$$\mathbf{R}_j(f) = \frac{1}{N}\sum_{n=1}^{N} \frac{1}{v_j(f,n)}\widehat{\mathbf{R}}_{\mathbf{c}_j}(f,n) \tag{10}$$

$$\mathbf{W}_j^{\mathrm{x}} = \mathbf{W}_j^{\mathrm{x}} \odot \frac{[\widehat{\mathbf{\Xi}}_j \odot \mathbf{V}_j^{\mathrm{x}}.^{-2} \odot \mathbf{V}_j^{\mathrm{f}}.^{-1}](\mathbf{U}_j^{\mathrm{x}}\mathbf{G}_j^{\mathrm{x}}\mathbf{H}_j^{\mathrm{x}})^T}{\mathbf{V}_j^{\mathrm{x}}.^{-1}(\mathbf{U}_j^{\mathrm{x}}\mathbf{G}_j^{\mathrm{x}}\mathbf{H}_j^{\mathrm{x}})^T} \tag{11}$$

$$\mathbf{U}_j^{\mathrm{x}} = \mathbf{U}_j^{\mathrm{x}} \odot \frac{\mathbf{W}_j^{\mathrm{x}T}[\widehat{\mathbf{\Xi}}_j \odot \mathbf{V}_j^{\mathrm{x}}.^{-2} \odot \mathbf{V}_j^{\mathrm{f}}.^{-1}](\mathbf{G}_j^{\mathrm{x}}\mathbf{H}_j^{\mathrm{x}})^T}{\mathbf{W}_j^{\mathrm{x}T}\mathbf{V}_j^{\mathrm{x}}.^{-1}(\mathbf{G}_j^{\mathrm{x}}\mathbf{H}_j^{\mathrm{x}})^T} \tag{12}$$

$$\mathbf{G}_j^{\mathrm{x}} = \mathbf{G}_j^{\mathrm{x}} \odot \frac{(\mathbf{W}_j^{\mathrm{x}}\mathbf{U}_j^{\mathrm{x}})^T[\widehat{\mathbf{\Xi}}_j \odot \mathbf{V}_j^{\mathrm{x}}.^{-2} \odot \mathbf{V}_j^{\mathrm{f}}.^{-1}]\mathbf{H}_j^{\mathrm{x}T}}{(\mathbf{W}_j^{\mathrm{x}}\mathbf{U}_j^{\mathrm{x}})^T\mathbf{V}_j^{\mathrm{x}}.^{-1}\mathbf{H}_j^{\mathrm{x}T}} \tag{13}$$

$$\mathbf{H}_j^{\mathrm{x}} = \mathbf{H}_j^{\mathrm{x}} \odot \frac{(\mathbf{W}_j^{\mathrm{x}}\mathbf{U}_j^{\mathrm{x}}\mathbf{G}_j^{\mathrm{x}})^T[\widehat{\mathbf{\Xi}}_j \odot \mathbf{V}_j^{\mathrm{x}}.^{-2} \odot \mathbf{V}_j^{\mathrm{f}}.^{-1}]}{(\mathbf{W}_j^{\mathrm{x}}\mathbf{U}_j^{\mathrm{x}}\mathbf{G}_j^{\mathrm{x}})^T\mathbf{V}_j^{\mathrm{x}}.^{-1}} \tag{14}$$

where $.^p$ denotes entrywise raising to the power p, N is the number of time frames in the STFT of the signal and $\widehat{\mathbf{\Xi}}_j = [\widehat{\xi}_j(f,n)]_{f,n}$, with

$$\widehat{\xi}_j(f,n) = \frac{1}{I}\operatorname{tr}(\mathbf{R}_j^{-1}(f)\widehat{\mathbf{R}}_{\mathbf{c}_j}(f,n)). \tag{15}$$

$\mathbf{W}_j^{\mathrm{f}}$, $\mathbf{U}_j^{\mathrm{f}}$, $\mathbf{G}_j^{\mathrm{f}}$ and $\mathbf{H}_j^{\mathrm{f}}$ are updated in a similar way.

After each EM iteration, the model parameters are normalised: the mean of $\mathbf{R}_j$, $\mathbf{W}_j^{\mathrm{x}}$, $\mathbf{U}_j^{\mathrm{x}}$, $\mathbf{G}_j^{\mathrm{x}}$, $\mathbf{H}_j^{\mathrm{x}}$, $\mathbf{W}_j^{\mathrm{f}}$, $\mathbf{U}_j^{\mathrm{f}}$ and $\mathbf{H}_j^{\mathrm{f}}$ are normalised to 1 while $\mathbf{G}_j^{\mathrm{f}}$ is multiplied by the product of the normalisation factors of the other variables.

The separated sources are then obtained via

$$\widehat{\mathbf{c}}_j(f,n) = \mathbf{\Omega}_j(f,n)\mathbf{x}(f,n). \tag{16}$$

3 Online EM-MU algorithm

We now consider an online context where in each time frame t, the data is limited to a block of M STFT frames indexed by n with $t - M + 1 \leq n \leq t$, where $M = 1$ for the stepwise approach and $M = N$ for the full offline approach. We define a step size coefficient $\alpha \in]0; 1]$ to stabilise the parameter updates by averaging over time. For each block, the spatial covariance matrices $\mathbf{R}_j^{(t)}(f)$ are initialised to a diffuse spatial covariance spanning a part of the audio space. The temporal weights $\mathbf{G}_j^{\mathrm{x}(t)}$ are randomly initialised and the normalised to the mean

spectral power of the signal. Finally, the temporal patterns $\mathbf{H}_j^{\mathrm{x}(t)}$ are initialised to diagonal matrices. The expectation of the natural statistics is computed using (8) and (9) for $t-M+1 \leq n \leq t$, whilst the spatial covariance matrix is updated as follows:

$$\mathbf{R}_j^{(t)}(f) = (1-\alpha)\mathbf{R}_j^{(t-1)}(f) + \alpha \left(\frac{1}{M} \sum_{n=t-M+1}^{t} \frac{1}{v_j(f,n)} \widehat{\mathbf{R}}_{\mathbf{c}_j}(f,n) \right) \quad (17)$$

where the superscript $^{(t)}$ denotes is the value of matrix for the block t.

$\mathbf{G}_j^{\mathrm{x}(t)}$ and $\mathbf{H}_j^{\mathrm{x}(t)}$ are updated using (13) and (14) for $t-M+1 \leq n \leq t$, as they are expected to significantly vary between blocks, whereas the updates of $\mathbf{W}_j^{\mathrm{x}}$ and $\mathbf{U}_j^{\mathrm{x}}$ become

$$\mathbf{W}_j^{\mathrm{x}(t)} = \mathbf{W}_j^{\mathrm{x}(t)} \odot \frac{\mathbf{M}_j^{\mathrm{x}(t)}}{\mathbf{C}_j^{\mathrm{x}(t)}} \quad (18)$$

$$\mathbf{U}_j^{\mathrm{x}(t)} = \mathbf{U}_j^{\mathrm{x}(t)} \odot \frac{\mathbf{N}_j^{\mathrm{x}(t)}}{\mathbf{D}_j^{\mathrm{x}(t)}} \quad (19)$$

where

$$\mathbf{M}_j^{\mathrm{x}(t)} = (1-\alpha)\mathbf{M}_j^{\mathrm{x}(t-1)} + \alpha[\widehat{\mathbf{\Xi}}_j \odot \mathbf{V}_j^{\mathrm{x}}.^{-2} \odot \mathbf{V}_j^{\mathrm{f}}.^{-1}](\mathbf{U}_j^{\mathrm{x}(t)}\mathbf{G}_j^{\mathrm{x}(t)}\mathbf{H}_j^{\mathrm{x}(t)})^T \quad (20)$$

$$\mathbf{C}_j^{\mathrm{x}(t)} = (1-\alpha)\mathbf{C}_j^{\mathrm{x}(t-1)} + \alpha\mathbf{V}_j^{\mathrm{x}}.^{-1}(\mathbf{U}_j^{\mathrm{x}(t)}\mathbf{G}_j^{\mathrm{x}(t)}\mathbf{H}_j^{\mathrm{x}(t)})^T \quad (21)$$

$$\mathbf{N}_j^{\mathrm{x}(t)} = (1-\alpha)\mathbf{N}_j^{\mathrm{x}(t-1)} + \alpha\mathbf{W}_j^{\mathrm{x}(t)T}[\widehat{\mathbf{\Xi}}_j \odot \mathbf{V}_j^{\mathrm{x}}.^{-2} \odot \mathbf{V}_j^{\mathrm{f}}.^{-1}](\mathbf{G}_j^{\mathrm{x}(t)}\mathbf{H}_j^{\mathrm{x}(t)})^T \quad (22)$$

$$\mathbf{D}_j^{\mathrm{x}(t)} = (1-\alpha)\mathbf{D}_j^{\mathrm{x}(t-1)} + \alpha\mathbf{W}_j^{\mathrm{x}(t)T}\mathbf{V}_j^{\mathrm{x}}.^{-1}(\mathbf{G}_j^{\mathrm{x}(t)}\mathbf{H}_j^{\mathrm{x}(t)})^T \quad (23)$$

where $\widehat{\mathbf{\Xi}}_j^{(t)}$ is computed as in (16). $\mathbf{M}_j^{\mathrm{f}(t)}$, $\mathbf{C}_j^{\mathrm{f}(t)}$, $\mathbf{N}_j^{\mathrm{f}(t)}$ and $\mathbf{D}_j^{\mathrm{f}(t)}$ are updated in a similar way. At each block, several iterations can be performed in order to improve the estimation of the model parameters.

Although equations (17) to (19) look similar to the online update of the local Gaussian model in [6] and [8], there are two crucial differences:

- The framework introduced in the current paper is more general in the sense that it uses hierarchical NMF, enabling the user to apply more specific constraints than when using shallow NMF.
- It is not limited to the sole use of the latest audio frame.

4 Experimental results

We compared the performance of the online audio source separation framework to the offline framework introduced in section 2.2, as a function of the number of EM iterations, α and M. The project aiming at remixing of recordings for sound engineers, DJs and consumers, we processed five 10 s long stereo commercial pop recordings composed of bass, drums, guitars, strings and voice. All the

recordings were recorded at 44100 Hz. The STFT was computed using half-overlapping 2048 sample sine windows. In the offline algorithm as well as in the online algorithm, each of the modeled sources were constrained in a way similar to section V.C in [9]. In the case of an harmonic source, $\mathbf{W}_j^{\mathrm{x}(t)}$ was fixed to a set of narrowband harmonic spectral patterns and the spectral envelope weights in $\mathbf{U}_j^{\mathrm{x}(t)}$ were updated, whereas for bass and percussive sources, $\mathbf{W}_j^{\mathrm{x}(t)}$ was a fixed diagonal matrix and $\mathbf{U}_j^{\mathrm{x}(t)}$ was a fixed matrix of basis spectra learned over a corpus of bass and drum sounds.

Audio samples of the separated sounds of this experiment can be found on http://www.irisa.fr/metiss/lssimon/LVA2012/index.html .

Separation performance was evaluated using the Signal-to-Distortion Ratio (SDR), the Signal-to-Interference Ratio (SIR), the source Image to Spatial distortion Ratio (ISR) and the Source-to-Artifacts Ratio (SAR) defined in [11]. For each set of conditions over the number of iterations, M and α, each of these criteria was averaged over all the mixtures and all the separated sound sources. Over all the results of this experiment, the SDR varied between -1.1 and 0.9 dB, the SIR between -4 and 1 dB, the ISR between 2.3 and 3.9 dB and the SAR between 10 and 19 dB.

Table 1. Separation performance (dB) of the offline and best online algorithms.

Algorithm	α	M	number of iterations	SDR	SIR	ISR	SAR
offline	N/A	N/A	100	0.8586	1.2837	3.7989	13.3872
online	1	50	30	**0.8671**	1.0675	**3.9690**	12.3278

As shown in table 1, when $\alpha = 1$, $M = 50$ and 30 GEM iterations are performed, the separation performance of the online algorithm is close to that of the offline algorithm. For smaller block size and smaller number of iterations, the performance decreases. For example, for $M = 10$ and 6 GEM iteration, the SDR is 0.53 dB and the SIR is 3.53 dB. More generally, fig. 1 shows that for $\alpha = 1$, increasing either the block size or the number of iterations increases the SDR, though the block size has less effect on the SDR than the number of iterations. The results also show that increasing the number of iterations from 10 to 30 increases the SDR by 0.2 dB, which can be considered as a significant improvement.

When $\alpha < 1$, the SDR decreases significantly as can be seen in fig. 1. It can also be seen that increasing the number of iterations decreases the SDR and changes of block size have little to no effect on the SDR. This can be explained by an inaccurate estimation of the model parameters of certain sources in the time intervals when these sources are inactive. These inaccurate parameters are then carried over subsequent time frames and may not converge back to accurate values. This undesirable effect is particularly salient for those parameters that are less constrained. For instance, with the considered model, the spatial covariance matrices of all sources gradually diverge towards a diffuse spatial covariance

spanning all directions in the mixture, while the effect is more limited for spectral parameters which are fixed or heavily constrained. Potential solutions to this problem are presented in the conclusion.

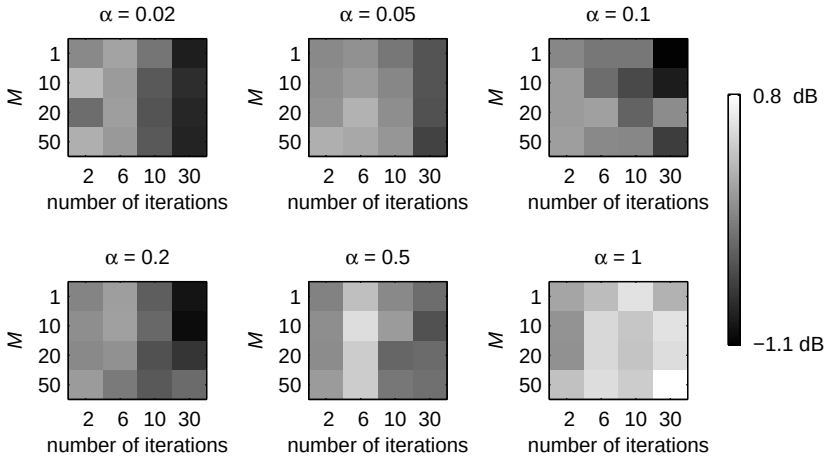

Fig. 1. Mean SDR for all sources and all mixtures, as a function of α, M and step size.

5 Conclusion

In this paper, a new framework for online audio source separation was presented. This algorithm offers an increased flexibility both in terms of the range of constraints that can be specified for each source and of the choice of a trade-off between separation accuracy and computational cost. It was shown that the separation accuracy is higher when the block size is large, but that small block sizes nevertheless offer an acceptable separation. However, small step sizes cause the spatial covariance matrices to diverge due to the presence of silence intervals in the sources.

This issue is well-known in the beamforming literature where a voice activity detector is used to restrict the time frames in which the model parameters are updated [12]. While this solution does not readily extend to source separation, we believe that there exist a number of alternative promising solutions, e.g. adding soft constraints over the least constrained parameters by means of probabilistic priors, using different step sizes for the most constrained and the least constrained parameters, and using signal-dependent step sizes related to

the power of $\mathbf{R}_{\mathbf{c}_j}(f,n)$ such that the parameters are not updated in the time intervals with low power.

Future work should also include an optimisation of the initialisation of the model parameters for each new block. After these improvements, we expect that the proposed framework will reach its full potential and provide a better trade-off between separation performance and computational cost.

Acknowledgements This work was supported by the EUREKA Eurostars i3DMusic project funded by Oseo.

References

1. Makino, S., Lee, T.-W. and Sawada, H.: Blind Speech Separation. Springer (2007)
2. Vincent, E., Jafari, M. G., Abdallah, S. A., Plumbley, M. D. and Davies, M. E.: Probabilistic modeling paradigms for audio source separation. In: Machine Audition: Principles, Algorithms and Systems. IGI Global, pp. 162–185 (2010)
3. Mukai, R., Sawada, H., Araki, S. and Makino, S.: Real-time Blind Source Separation For Moving Speakers Using Blockwise ICA and Residual Crosstalk Subtraction. In: 4th Int. Symp. Independent Component Analysis and Blind Signal Separation, pp. 975–980 (2003)
4. Mori, Y., Saruwatari, H., Takatani, T., Ukai, S., Shikano, K., Hiekata, T., Ikeda, Y., Hashimoto, H. and Morita, T.: Blind separation of acoustic signals combining SIMO-model-based independent component analysis and binary masking. EURASIP Journal on Advances in Signal Processing, vol. 2006, issue 1, pp. 1–17 (2006)
5. Loesch, B. and Yang, B.: Online blind source separation based on time-frequency sparseness. In: Proc. 2009 IEEE Int. Conf. on Acoustics, Speech and Signal Processing, pp. 117–120 (2009)
6. Togami, M.: Online speech source separation based on maximum likelihood of local Gaussian modeling. In: Proc. 2011 IEEE Int. Conf. on Acoustics, Speech and Signal Processing, pp. 213–216 (2011)
7. Ono, N., Miyamoto, K. and Sagayama, S.: A real-time equalizer of harmonic and percussive components in music signals. In: Proc. 2008 Int. Conf. on Music Information Retrieval, pp. 139 – 144 (2008)
8. Wang, D, Vipperla, R. and Evans, N.: Online pattern learning for non-negative convolutive sparse coding. In: Proc. Interspeech'11, pp. 65–68 (2011)
9. Ozerov, A., Vincent, E. and Bimbot, F.: A general flexible framework for the handling of prior information in audio source separation. IEEE Transactions on Audio, Speech, and Language Processing, to appear
10. Duong, N.Q.K., Vincent, E. and Gribonval, R.: Under-determined reverberant audio source separation using a full-rank spatial covariance model. IEEE Transactions on Audio, Speech, and Language Processing, vol. 18, issue 7, pp. 1830–1840 (2010)
11. Vincent, E., Sawada, H., Bofill, P., Makino, S. and Rosca, J. P.: First stereo audio source separation evaluation campaign: data, algorithms and results. In: Proc. 2007 Int. Conf. on Independent Component Analysis and Blind Source Separation, pp. 552–559 (2007)
12. Brandstein, M.S., Ward, D.B.: Microphone Arrays: Signal Processing Techniques and Applications. Springer (2001)