Convolutive Audio Source Separation Using Robust ICA and Reduced Likelihood Ratio Jump

Dimitrios Mallis, Thomas Sgouros, Nikolaos Mitianoudis

Convolutive Audio Source Separation using Robust ICA and Reduced Likelihood Ratio Jump

Dimitrios Mallis, Thomas Sgouros, and Nikolaos Mitianoudis

Electrical and Computer Engineering Dep., Democritus University of Thrace, Xanthi 67100, Greece,
malldimi1@gmail.com, tsgouros@ee.duth.gr, nmitiano@ee.duth.gr,

Abstract. Audio source separation is the task of isolating sound sources that are active simultaneously in a room captured by a set of microphones. Convolutive audio source separation of equal number of sources and microphones has a number of shortcomings including the complexity of frequency-domain ICA, the permutation ambiguity and the problem's scalabity with increasing number of sensors. In this paper, the authors propose a multiple-microphone audio source separation algorithm based on a previous work of Mitianoudis and Davies [1]. Complex FastICA is substituted by Robust ICA increasing robustness and performance. Permutation ambiguity is solved using the Likelihood Ration Jump solution, which is now modified to decrease computational complexity in the case of multiple microphones.

1 Introduction

The problem of Blind Audio Source Separation (BASS) implies the extraction of independent audio sources from an audio mixture that has been observed by a number of microphones, without any prior knowledge regarding the involved sources or the mixing system. In recent years, many methods have been proposed for resolving this problem with relative success. BASS becomes more complicated when we are dealing with real-room audio recordings. In reverberant rooms, each source is recorded multiple times by each microphone under different time delays and amplifications, due to sound waves' reflections on the room surfaces. The mixing system can thus be modelled using a room impulse response of finite length (FIR filter). In the general case of M microphones that capture a mixture of N sources, a common representation of the aforementioned mixing is $\boldsymbol{x}(t) = \boldsymbol{A} * \boldsymbol{s}(t)$, where $*$ denotes linear convolution, $\boldsymbol{s}(n) = [s_1(n), s_2(n), ..., s_N(n)]^T$ are the source signals, $\boldsymbol{x}(n) = [x_1(n), x_2(n), ..., x_M(n)]^T$ are the observation signals, n is the time index and $\boldsymbol{A}$ is a matrix, whose elements $\boldsymbol{a}_{ij}$ are FIR filters, describing the room impulse responses between the j-th source and the i-th microphone.

A classic decomposition method for performing BASS is Independent Component Analysis (ICA). ICA extracts Independent Components (ICs) from a linear,

instantaneous mixture, assuming independence between the original sources. In real rooms, where we deal with convolutive mixtures, ICA can also be applied by moving the separation to the frequency domain, where the convolution between the sources and the room transfer function is reduced to multiplication [2] for a number of discrete frequency bins L i.e. $\boldsymbol{x}(f,n) = \boldsymbol{A}_f \boldsymbol{s}(f,n)$, where $f = 1, \ldots, L$. In other words, we transform a difficult convolution problem to a number of easier instantaneous problems, that can be solved using ICA.

By solving the separation problem in the frequency domain, ICA introduces 2 ambiguities: scale and permutation. The first results into random scaling of the extracted ICs, which can cause spectral deformations and reduce separation quality. The latter results into arbitrary source permutations along the discrete frequency bins and as a result inability to achieve separation. The scale ambiguity can be resolved easily as a post processing step, by mapping the estimated sources back to the microphones' domain and recover the signals as they have been originally observed by the microphones [1].

The permutation ambiguity, on the other hand, is a difficult problem, and various techniques have been proposed without featuring robust performance in all cases. In [3], Mazur and Mertins align permutations by using generalised Gaussian Distribution in order to find differences between neighbouring frequency bins. Sawada et al. [4] exploit the correlation coefficients of amplitude envelopes, which if maximised show the correct source alignment, while in [5] Saito et al. utilise, for the same purpose, the correlation between interfrequency power ratios. A different approach was followed by Sarmiento et al. [6] who find the spectral similarities between the separated components in the frequency domain by employing a contrast function. A region-growing approach, to minimise the spreading of possible misalignments, in order to improve permutation alignment, was introduced by Wang et al. [7].

Most of the available methods for tackling the convolutive source separation problem are focused on the two-source two-microphone (2×2) case. However, low-cost commercially available hardware, such as the Microsoft Kinect interface, has been developed to offer low-latency four-microphone recordings and can be used to process 4×4 cases. In this paper, we focus on the problem of audio source separation for determined cases (equal number of microphones and sources) that involves more than two sources. The presented methodology offers a computationally efficient solution for both the separation task as well as the permutation ambiguity. Based on the Kinect interface, we created a set of recordings containing mixtures of multiple sources as well as the original sources for evaluation purposes. This dataset is publicly available for further evaluation of audio separation methods[1]. In this dataset, we will apply a novel framework that is optimized for multiple sources. This is then compared with the previous work of Mitianoudis and Davies [1] to observe its efficiency for multiple sources. The proposed framework includes a robust complex ICA separation algorithm, called RobustICA [8], that has not been used before for convolutive audio source separation. In addition, we present a new technique to tackle the permutation

[1] Dataset available at `http://utopia.duth.gr/nmitiano/download.html`

ambiguity problem, especially for large number of sources, based on the Likelihood Ratio Jump solution [1]. We show that this new technique can reduce the computational cost of addressing the permutation problem in comparison to the original Likehood Ratio Jump and can produce the same, if not better separation quality.

2 Instantaneous Complex Source Separation

In the instantaneous case, we consider the following mixing process: $\boldsymbol{x}(n) = \boldsymbol{A} \cdot \boldsymbol{s}(n)$. In order to separate the sources, we have to estimate an unmixing matrix $\boldsymbol{W}$, such that $\boldsymbol{u}(n) = \boldsymbol{W} \cdot \boldsymbol{x}(n) \approx \boldsymbol{s}(n)$.

2.1 The FastICA algorithm

In the determined case, where the number of sources is equal to the number of observations ($N = M$), the most popular method of estimating the unmixing matrix $\boldsymbol{W} = [\boldsymbol{w_1}, \boldsymbol{w_2}, ..., \boldsymbol{w_N}]^T$ is the FastICA algorithm. There are many implementations of the FastICA algorithm, that are based on optimization of a contrast function emphasizing nonGaussianity using a fixed-point iteration algorithm. One common fixed point algorithm is the following [9],

$$\Delta \boldsymbol{W}_f = \boldsymbol{D}[\mathrm{diag}(-\alpha_i) + \mathcal{E}\{\phi(\boldsymbol{u})\boldsymbol{u^H}\}]\boldsymbol{W_f} \tag{1}$$

where $\phi(u) = u/|u|$ is an activation function for superGaussian sources, α_i an adaptive parameter and $\mathcal{E}\{\cdot\}$ denotes the expectation operator. This iterative update of the unmixing matrix $\Delta \boldsymbol{W}_f$ is calculated using a maximum likelihood estimator. The method also needs the data of every frequency bin to be prewhitenned. Even though this method has been initially introduced for real-data mixture, it has shown to work well with complex data in [1].

2.2 The RobustICA algorithm

In this section, we examine a source separation algorithm, named RobustICA [8]. RobustICA optimizes the following generalized form of kurtosis.

$$\mathcal{K}(\boldsymbol{w}) = \frac{\mathcal{E}\{|y|^4\} - 2\mathcal{E}^2\{|y|^2\} - |\mathcal{E}\{y^2\}|^2}{\mathcal{E}^2\{|y|^2\}} \tag{2}$$

The above definition of kurtosis can be applied to both real and complex data. In addition, prewhitening is not necessary for RobustICA. RobustICA uses exact line search optimization of the absolute kurtosis contrast function, instead of fixed-point optimization, used by FastICA [10].

$$\mu_{opt} = \arg\max_{\mu}(\mathcal{K}(\boldsymbol{w} + \mu \boldsymbol{g})) \tag{3}$$

The search direction can be given by the gradient of the kurtosis $\boldsymbol{g} = \boldsymbol{\nabla_w}\mathcal{K}(\boldsymbol{w})$. Exact line search is often a computationally expensive optimization technique

that requires additional numerical analysis algorithms. In the case of kurtosis, the optimal step size μ_{opt} is calculated algebraically with a minimum computational cost. It is shown in [8] that μ_{opt} can be calculated from the root of a low-degree polynomial that maximizes the absolute value of the contrast function along the search direction. RobustICA has a number of advantages [8] compared to the original FastICA:

- RobustICA does not make any assumption regarding the the sources' statistical profile, and can deal with real and complex sources alike.
- Prewhitening is not mandatory before RobustICA. Multiple ICs in that case can be extracted with the method of linear regression in contrast to symmetric orthogonalization that is used by FastICA.
- The method can target sub-Gaussian or super-Gaussian sources in a specific order. This feature is useful in the audio separation case, where we know in advance that data in the frequency domain can be mostly modelled as super-Gaussian [1].
- The method is robust to the presence of saddle points and spurious local extrema of the contrast function [8].
- RobustICA can achieve great separation performance with relatively small additional computational cost, compared to other ICA implementations. This feature is demonstrated in [8] and will be verified by the experimental results in this paper.

Despite the fact that prewhitening is not mandatory for RobustICA, it will be used as a preprocessing step in our proposed framework. This is due to the observation that it leads to a more computationally efficient implementation in the case of multiple sources. Since the prewhitened components lay on an orthogonal structure, every ICA iteration that attracts one IC towards an original source, forces the rest of the ICs to converge faster to other sources. This can be achieved with the use of symmetric orthogonalization, as in (4).

$$\boldsymbol{W}_f^{+} \leftarrow \boldsymbol{W}_f(\boldsymbol{W}_f^H \boldsymbol{W}_f)^{-0.5} \tag{4}$$

On the contrary, in linear regression, after the extraction of an IC, we have to separate a reduced mixture from a random position, which can be rather slow. As a result, we use prewhitening to improve the convergence speed of our method, in expense of the separation performance limitations that prewhitening can introduce.

3 Frequency-domain source separation

Frequency-domain source separation methods apply the Short-Time Fourier Transform (STFT) to the mixture recordings $\boldsymbol{x}(t)$. Consequently, the convolutive mixture is transformed to L instantaneous mixture via the STFT, i.e. $\boldsymbol{x}(t) = \boldsymbol{A} * \boldsymbol{s}(t) \Rightarrow \boldsymbol{X}(f,t) = \boldsymbol{A}_f \boldsymbol{S}(f,t)$. The separation problem can be solved independently using any complex ICA algorithm, such as the RobustICA. ICA's inherent

scale and permutation ambiguities impose severe problems in this framework and must be resolved. Scale ambiguity is tackled using a mapping to the microphone domain [1]. There exist many methods to tackle the permutation ambiguity of frequency-domain BASS methods.

3.1 Likelihood Ratio Jump

Mitianoudis and Davies introduced in [1] the Likelihood Ratio Jump method for the alignment of frequency bins to the correct source. This method can be used either after each iteration of the ICA algorithm, or even better as a post-processing mechanism. The method works iteratively and in each iteration forms a likelihood ratio test to decide, which permutation is the most probable for each frequency bin. It uses a set of rescaling parameters γ_{ij} that model the probability that the i^{th} source has moved to the j^{th} position. For each frequency bin, it calculates the probabilities for all the possible permutations. For example, in a mixing of 3 sources a possible permutation of the extracted ICs: $IC3 \rightarrow IC1$, $IC1 \rightarrow IC2$, $IC2 \rightarrow IC3$, forms the probability:

$$L = -\log(\gamma_{31}\gamma_{12}\gamma_{23}) \tag{5}$$

The correct permutation is the one that produces the maximum probability as in (5). For the case of three sources, there are $3! = 6$ possible permutations for the extracted ICs, which have to assessed probabilistically, to conclude which permutation is the most likely to be correct. The parameters γ_{ij} are produced through a maximum likelihood estimator and can be calculated as follows:

$$\gamma_{ij} = \frac{1}{T}\sum_{t}\frac{|u_i(f,t)|}{\beta_j(t)} \tag{6}$$

where $u_i(f,t)$ is the value of IC i for the discrete frequency bin f and time index t and β_j is a non-stationary time-varying scale parameter that is calculated for the source j. Finally, T is the number of observations.

The parameter $\beta_j(t)$ incorporates information related to the signal's spectral envelope over time, thus it can be interpreted as a volume measurement. Literally, it measures the overall signal amplitude along the frequency axis, emphasizing the fact that one source is "louder" than others at a certain time slot. This "temporal energy burst" can force alignment of the permutations along the frequency axis. A possible estimation for the β_j parameter can be the following:

$$\beta_j = \frac{1}{L}\sum_{f}|u_j(f,t)| \tag{7}$$

where L is the number of frequency bins. The Likelihood Ratio Jump (LRJ) method has demonstrated very stable performance in solving the permutation ambiguity for a large number of cases [1]. However, this was mainly demonstrated for 2×2 cases.

3.2 Reduced Likelihood Ratio Jump

One major disadvantage of this method is its computational cost that increases rapidly with the number of sources, as for each iteration of the algorithm we need to make $N!$ comparisons. For example, we can consider a case of 5 sources, where the FFT has 4096 frequency bins, and the post-processing permutation method needs to spend 15 iterations for the system to converge to the correct permutation for most of the bins. In total, we will need $5! \times 2048 \times 15 = 125 \times 4096 \times 15$ calculations of the expressions in (5) and as a result the whole task is computationally inefficient, if not prohibitive.

In this section, we propose a new "suboptimal" method, named Reduced Likelihood Ratio Jump. This technique selects to perform a few major comparisons in contrast to full set of $N!$ comparisons in the original method, thus the term "suboptimal". Nonetheless, we witnessed that it can produce the same, if not better separation quality with a considerable reduction of the computational cost.

The Reduced Likelihood Ratio Jump is based on the iterative nature of the original method. The Likelihood Ratio Jump needs, in most of the examined examples, some dozen iterations for every frequency bin to converge to the correct permutation. This is mainly due to the parameter $\beta(t)$. As previously mentioned, this parameter incorporates information about the time envelope of the signal. As more permutations are sorted in each iteration, the time envelope of each signal becomes more distinct and as a result, the parameter $\beta(t)$ has a stronger impact in the calculation, that helps resolving the permutation for frequency bins, where this task is more difficult.

During extensive experimentation, we witnessed that there are many frequency bins that feature correct permutation from the first or second iteration of the method. For the remaining frequency bins, the algorithm after some iterations needs to swap only one IC pair to restore the correct permutation, since the others have already been sorted to the correct sources. This situation is common for cases with many sources, as for most of the ICs the correct permutation is clear after a small number of iterations, and only one or two pairs may need an improved calculation of the parameter $\beta(t)$ to be permuted correctly.

Based on the above observation, we propose to reduce the number of examined permutations that our method considers in every iteration of the algorithm. As the method works iteratively, we propose to calculate the most probable permutation from a set permutation that only includes the swapping of one pair of ICs at a time. In the case of N sources, in one iteration we can calculate the most probable from $N-1$ swaps between the ICs, as happens in Likelihood Ratio Jump, but with the progression of the method only one swapping will be needed to ensure the correct permutation of the sources. Even if more than one pairs of ICs are permuted incorrectly, as the method progress the correct permutation will be restored, one pair at a time. As a result, we employ the iterations needed to make a more accurate estimation of the parameter $\beta(t)$, to reduce the set of examined permutations and consequently the required computational time. By examining only one pair of permutations at a time, we reduce the complexity of

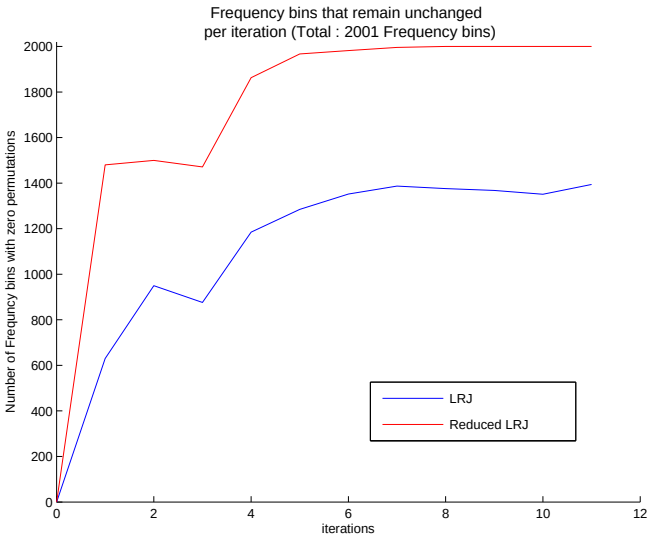

Fig. 1. Comparison between Reduced LRJ and original LRJ. More frequency bins remained unchanged using the Reduced LRJ for the same number of iterations.

the method from $N!$ to $\frac{1}{2}N(N-1)+1$, which for 5 sources means a reduction from 120 to 11 comparisons per iteration. As we will show in the experimental section this suboptimal method does not undermine the quality of the separation but may also enhance it instead.

In Fig. 1, we can see that the Reduced Likelihood Ratio Jump converges for more frequency bins, compared to the original method for the same number of iterations. In an example of a total of 2001 frequency bins the Reduced LRJ has concluded the permutation sorting of all the frequency bins from the 8th iteration, in contrast to the original method that always needs to sort about 600 more frequency bins. For some bins, the original LRJ changes permutation in every iteration. This phenomenon is due to the very small differences between the likelihood values that force the original method to toggle between 2 permutations for specific frequency bins.

4 Experiments

4.1 Evaluation process

To evaluate the performance of our proposed framework, we created an evaluation dataset of 7 audio recordings. This evaluation dataset contains 5 mixtures of 3 sources and 2 mixtures of 4 sources, featuring both speech and music. The recordings were made using the Microsoft Kinect interface. The sources-microphones were placed in different positions in a real reverberant room. It also includes separate, recordings of the corresponding sources under the same

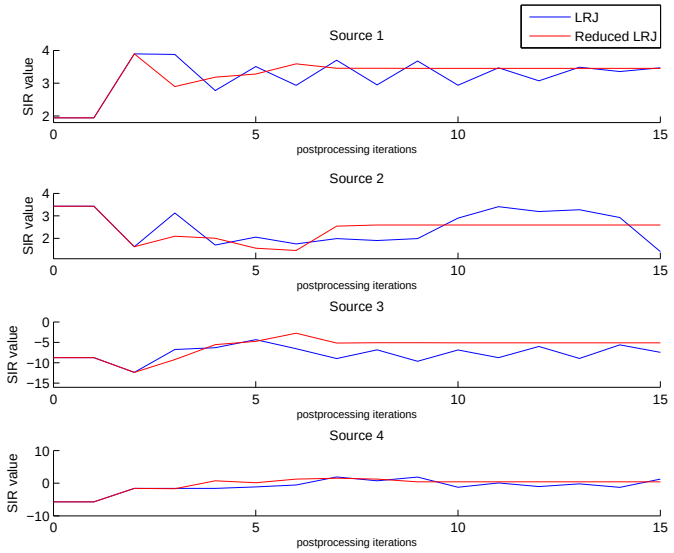

Fig. 2. Separation Quality in terms of SIR for the LRJ and the Reduced LRJ methods.

recording conditions (position and loudness) in order to be used as ground truth for the evaluation of the separation quality achieved by the separation framework[2].

We used the RobustICA MATLAB implementation, as provided freely by the authors[3], and we made the necessary modifications to work in a convolutive source separation framework with original and the Reduced LRJ. For the experiments presented in this section, we assume a room impulse response of 90.7 ms length, which we reckon is a valid model for the recording positions in the reverberant room. To measure the separation quality, we used the metrics SDR, SIR and SAR that are designed specifically for Performance Measurement in Blind Audio Source Separation [11]. These metrics are implemented as a publicly-available MATLAB Toolbox and were designed to allow a time-invariant filter distortion of 512 samples length. We manually changed this value to 2000 samples, to cater for the wrong synchronization between the original and the estimated sources that are extracted from the mixture [11].

Metrics SDR and SAR measure the distortion and artifacts that are created by the separation method. Both examined frameworks (the proposed framework and Mitianoudis-Davies) produce similar values as they employ similar contrast functions with different optimization methods. To avoid the repetition of similar experimental results, we will therefore present SIR measurements only, and the separation quality in terms of interference elimination between the extracted sources.

[2] Dataset available from `http://utopia.duth.gr/nmitiano/download.html`

[3] `http://www.i3s.unice.fr/~zarzoso/robustica.html`

Table 1. Efficiency Comparison between the Frequency Domain RobustICA and FastICA implementations in terms of Signal-Interference-Ratio (dB). Here, we compare the performance of 3 and 15 iterations of RobustICA and FastICA. RobustICA reaches a better separation result faster than FastICA.

Recording	**Source**	3 iterations		15 iterations	
		RobustICA	FastICA	RobustICA	FastICA
1	1	-0.464	-0.09	-0.09	0.88
	2	1.698	0.99	0.60	1.51
	3	-7.30	-7.98	-8.84	-7.99
2	1	-6.36	-4.25	-6.79	-3.37
	2	5.53	0.23	3.46	1.07
	3	1.12	-2.00	0.67	-0.38
3	1	5.40	3.20	6.44	4.20
	2	0.18	-1.17	-0.63	-2.38
	3	-10.11	-8.98	-10.31	-9.78
4	1	-2.40	-2.16	-3.34	-0.10
	2	2.26	1.75	2.82	3.53
	3	-6.78	-6.22	-6.24	-10.09
5	1	3.81	-2.47	4.12	1.12
	2	-5.78	-6.05	-6.61	-8.02
	3	4.01	0.99	4.37	3.05
6	1	-6.80	-1.33	-6.79	-4.02
	2	-3.78	-6.82‘	-4.57	-3.72
	3	1.18	-0.05	1.73	4.27
	4	-0.63	-2.17	-0.95	-1.50
7	1	2.44	1.19	0.50	0.64
	2	-1.68	0.30	-0.25	-0.77
	3	-6.99	-7.68	-6.72	-7.87
	4	-7.83	-3.38	-5.17	-5.02

4.2 Performance comparison

In this section, we present several experiments to demonstrate the efficiency of the 2 examined frameworks using FastICA, RobustICA, the LRJ and the reduced LRJ. Firstly, in Table 1 we compare the separation quality of the two ICA implementations in terms of SIR. To tackle the permutation ambiguity, we use, in this experiment, 5 iterations of the original LRJ of Mitianoudis and Davies [1] for the two frameworks. We can see that:

Table 2. Comparison between the Likelihood Ratio Jump and the Reduced Likelihood Ratio Jump for 8 available iterations in terms of SIR (dB).

Method	Source	**rec 1**	**rec 2**	**rec 3**	**rec 4**	**rec 5**
LRJ	1	-0.24	-6.46	6.18	-1.55	3.82
	2	1.67	5.42	0.29	1.13	-5.78
	3	-6.98	1.03	-10.71	-6.49	4.01
Reduced LRJ	1	2.72	-2.82	6.44	-0.85	3.18
	2	1.58	3.82	1.91	1.46	-7.64
	3	-7.87	-2.57	-5.78	-6.49	2.41

Table 3. Running time comparison between the two frameworks (seconds)

Recording	framework 1			framework 2		
	separation	Perm	total	separation	Perm	total
1 (3 sources)	5.58	5.89	13.04	14.38	7.16	23.18
2 (3 sources)	4.89	5.21	11.43	11.30	6.00	18.61
3 (3 sources)	4.45	2.71	7.70	5.27	2.56	8.39
4 (3 sources)	4.63	2.96	8.20	5.92	2.92	9.48
5 (3 sources)	5.00	5.24	11.60	11.61	6.16	19.17
6 (4 sources)	6.73	8.93	17.36	22.25	17.86	41.82
7 (4 sources)	6.99	10.26	19.27	22.68	20.58	45.29

- As the 2 methods perform optimization of different criteria, they do not perform the same for the examined cases. RobustICA performs better for cases (2,3,5), FastICA for (4,6) and for the rest of the recordings we observe similar separation qualities. In general, we can say that for the examined cases RobustICA with prewhitenning, can reach and outperform slightly the original method of FastICA in separation quality.
- RobustICA presents very fast convergence. In all examined cases, it produces very good separation quality in only 3 iterations. This feature of RobustICA can be a great advantage, compared to previous ICA implementations, as also mentioned in [8]. In contrast, FastICA needs more iterations to produce stable results. We can see, in Table 1, the major differences in separation quality from 3 to 15 iterations of FastICA, in comparison to RobustICA that reaches very good separation quality in only 3 iterations, which is then only slightly improved as the iterations increase. Despite the fact that a RobustICA iteration is more costly than the FastICA equivalent, its fast convergence improves the total computational efficiency.

In the next experiment, we compare the separation quality with 8 iterations for both the new Reduced and the original Likelihood Ratio Jump method of Mitianoudis and Davies [1]. For the separation of frequency bins in this exper-

iment, we have used the RobustICA with prewhitenning as it has shown to be the most efficient method. In Table 2, we can see the separation performance produced by the 2 permutation solving methods for the 3-sources recordings. The LRJ perform better only in recording 2. For the rest of the recordings, the Reduced Likelihood Ratio performs better despite the fact that it is a suboptimal method.

This improved performance of our proposed method can be due to the stability that is produced from the convergence of a greater number of frequency bins, as shown previously in Fig. 2. The Reduced Likelihood Ratio Jump, by allowing a smaller set of possible swaps, leads to a more stable state for a larger number of frequency bins. In the separation example of recording 6 (4 sources), shown in Fig. 2, we observe that the separation quality, produced by the Reduced Likelihood Ratio Jump, is greater to the original method, for every extracted source. Due to the accurate convergence for larger number of frequency bins, the Reduced Likelihood Ratio Jump reaches a constant separation quality from a smaller number of iterations. In contrast, the separation quality using the original method seems to vary, depending on the permutation arising from the frequency bins that do not converge in every iteration.

Finally, in Table 3, we present a comparison of the running times required by the 2 frameworks to produce stable separation results. The computational times of Table 3 refer to MATLAB R2013 implementations of the examined frameworks on a Pentium i7 3.4GHz PC with 8 GB RAM. As explained previously, RobustICA requires sufficiently less iterations than FastICA and in the results of Table 1 we use 3 iterations for framework 1 (RobustICA) and 15 iterations for framework 2 (FastICA). For the permutation ambiguity, both the Reduced Likelihood Ratio Jump and the Likelihood Ratio Jump required 9 iterations. We used 9 iterations that seemed to be a good choice in Fig. 2 since after 9 iterations on average, all sources seems to present a relevant stability in the calculated SIR values. We can see that the improvement in computational time for the examined recordings is important, with similar separation performance as shown in previous experiments. RobustICA with much less iterations requires about 1/3 of the FastICA computational time, while the Reduced Likelihood Ratio Jump can solve the permutation ambiguity in about half the time of the original LRJ. As an example, for the seventh recording we need 19 sec with the proposed framework and 45 with the original one, which demonstrates the efficiency of the proposed approach.

5 Conclusion

In this paper, we presented an extension of the previous work of Mitianoudis and Davies for convolutive audio source separation. First of all, the FastICA separation algorithm was replaced with the RobustICA algorithm, which improves the performance and stability of the framework. The next improvement was the proposal of a Reduced LRJ solution, in order to reduce the increased computational cost in the case of more than 2 sensors and sources. The Reduced LRJ, although

a suboptimal solution, seems to achieve better separation, since it doesn't allow more source flippings than necessary. The new framework has been tested on a newly recorded dataset using the cost-efficient Microsoft Kinect platform with success. For future work, the authors would like to extend this framework for underdetermined recordings, i.e. dealing with real-room recordings containing more sources than sensors.

References

1. Mitianoudis, N., Davies, M.: Audio source separation of convolutive mixtures. IEEE Trans. on Speech and Audio Processing **11**(5) (Sep 2003) 489–497
2. Smaragdis, P.: Blind separation of convolved mixtures in the frequency domain. Neurocomputing **22**(1) (1998) 21–34
3. Mazur, R., Mertins, A.: An approach for solving the permutation problem of convolutive blind source separation based on statistical signal models. IEEE Trans. on Audio, Speech, and Language Processing **17**(1) (Jan 2009) 117 – 126
4. Sawada, H., Araki, S., Makino, S.: Underdetermined convolutive blind source separation via frequency bin-wise clustering and permutation alignment. IEEE Trans. on Audio, Speech, and Language Processing **19**(3) (Mar 2011) 516 – 527
5. Saito, S., Oishi, K., Furukawa, T.: Convolutive blind source separation using an iterative least-squares algorithm for non-orthogonal approximate joint diagonalization. IEEE Trans. on Audio, Speech, and Language Processing **23**(12) (Dec 2015) 2434 – 2448
6. Sarmiento, A., Duran-Diaz, I., Cichocki, A., Cruces, S.: A contrast function based on generalised divergences for solving the permutation problem in convolved speech mixtures. IEEE Trans. on Audio, Speech, and Language Processing **23**(11) (Nov 2015) 1713 – 1726
7. Wang, L., Ding, H., Yin, F.: A region-growing permutation alignment approach in frequency-domain blind source separation of speech mixtures. IEEE Trans. on Audio, Speech, and Language Processing **19**(3) (Mar 2011) 2434 – 2448
8. Zarzoso, V., Comon, P.: Robust independent component analysis by iterative maximization of the kurtosis contrast with algebraic optimal step size. IEEE Trans. on Neural Networks **21**(2) (Feb 2010) 248–261
9. Hyvärinen, A.: The fixed-point algorithm and maximum likelihood estimation for independent component analysis. Neural Processing Letters **10**(1) (1999) 1–5
10. Hyvärinen, A.: Fast and robust fixed-point algorithms for independent component analysis. IEEE Trans. on Neural Networks **10**(3) (1999) 626–634
11. Févotte, C., Gribonval, R., Vincent, E.: BSS EVAL Toolbox User Guide. Technical report, IRISA Technical Report 1706, Rennes, France, April 2005, http://www.irisa.fr/metiss/bss eval/