SEPARATION OF STEREO SPEECH SIGNALS BASED ON A SPARSE DICTIONARY ALGORITHM

Maria G. Jafari and Mark D. Plumbley

Centre for Digital Music
Queen Mary University of London.
Mile End Road, London E1 4NS, UK
Email: maria.jafari@elec.qmul.ac.uk; mark.plumbley@elec.qmul.ac.uk
Web: http://www.elec.qmul.ac.uk/digitalmusic/

ABSTRACT

We address the problem of source separation in echoic and anechoic environments, with a new algorithm which adaptively learns a set of sparse stereo dictionary elements, which are then clustered to identify the original sources. The atom pairs learned by the algorithm are found to capture information about the direction of arrival of the source signals, which allows to determine the clusters. A similar approach is also used here to extend the dictionary learning K singular value decomposition (K-SVD) algorithm, to address the source separation problem, and results from the two methods are compared. Computer simulations indicate that the proposed adaptive sparse stereo dictionary (ASSD) algorithm yields good performance in both anechoic and echoic environments.

1. INTRODUCTION

In recent years, sparse signal decompositions and dictionary learning techniques have often been applied to the problem of source separation, when dealing with mixtures arising from real environments [1]. Since the main underlying assumption with this type of techniques is the sparsity of the sources in some domain, they are sometimes collectively denoted as sparse component analysis (SCA) [2]. The aim of SCA is to solve the source separation problem by a multi-stage procedure, that typically involves the following four steps [2]. Firstly, apply a sparsifying transform to the observed data, in order to move to a domain where the sources are sparse. Secondly, estimate the mixing. This step often consists of clustering the transform coefficients based on techniques such as K-means, and it relies on the sources not overlapping (or almost not overlapping) in the transform domain. Thirdly, find the source representation using, for instance, binary masking. Fourthly, reconstruct the sources by inverting the sparsifying transform. The first step in the above procedure is of particular interest here. The signal is to be transformed into a domain where the sources are sparse, and therefore it is expected that, in this domain, the signal representations of the sources do not overlap [3]. Orthogonal sparsifying transforms such as the short time fourier transform (STFT) is often used in this stage, but fixed or learned overcomplete dictionaries are acquiring popularity.

We address the source separation problem according to the procedure described above, using a new adaptive sparse stereo dictionary learning algorithm in the first stage of SCA. The algorithm that we propose is based on a greedy approach, learning a dictionary of stereo atoms from both channels simultaneously [4]. The method maximizes the L2-norm of the data, re-arranged into frames, while minimizing its L1-norm, hence seeking dictionary elements that are sparse, as well as yielding sparse representations for the signals. Moreover, the transform is forced to be orthogonal by removing all the components lying in the direction of a particular vector, corresponding to the selected data frame, at each iteration. Thus, the inverse transform is evaluated via multiplication by the transpose of the dictionary matrix.
The K-SVD dictionary learning algorithm in [5] is also used, in this paper, to address the source separation problem, as part of the first stage of SCA, and its performance is compared to that of the proposed ASSD-based separation algorithm. The structure of the paper is as follows: the problem that we seek to address is outlined in Section 2, and our proposed sparse stereo adaptive dictionary algorithm for source separation is introduced in Section 3. K-SVD is described in Section 4, where it is also extended to source separation. Finally, experimental results are presented in Section 5, while conclusions are drawn in Section 6.

2. PROBLEM STATEMENT

The convolutive blind audio source separation problem arises when an array of microphones records mixtures of a set of sound sources $\mathbf{s}(n)$ that are convolved with the impulse response between each source and sensor. When 2 sources and 2 microphones are present, the signal recorded at the i-th microphone, $x_i(n)$, is

$$x_i(n) = \sum_{j=1}^{2} \sum_{l=1}^{L} a_{ij}(l) s_j(n-l), \quad i = 1,2 \tag{1}$$

where $s_j(n)$ is the i-th source signal, $a_{ij}(l)$ denotes the impulse response from source j to sensor i, and L is the maximum length of all impulse responses. The aim of source separation is to find estimates for the unmixing filters $w_{ij}(l)$, using only the sensor measurements, and to reconstruct the sources from

$$y_j(n) = \sum_{i=1}^{2} \sum_{l=1}^{L} w_{ij}(l) x_i(n-l), \quad j = 1,2 \tag{2}$$

where $y_j(n)$ is the j-th recovered source.

3. PROPOSED SEPARATION METHOD

In this paper, we address the source separation problem based on the SCA procedure, as follows:

1. Reshape the observed signal vector $\mathbf{x}(n) \in \mathbb{R}^{n_{\max}}$, into a matrix $\mathbf{X}$.
2. Apply a sparse dictionary learning algorithm to learn stereo atoms from the two mixtures.
3. Cluster the learned atom pairs.
4. Reconstruct the sources.

In [6] we used a gradient-based sparsifying independent component analysis (ICA) algorithm, but this was very slow, even on relatively small frame sized of 512 samples. We now replace this with a new, much faster, greedy algorithm to perform this transform.

The first step entails stacking samples pairs of the observed stereo data, as described in [6], and dividing the resulting data into blocks of overlapping data frames, resulting into the matrix $\mathbf{X}(n)$, containing the stereo mixture. Reshaping the data in this manner allows the modeling of both correlations between microphones, and correlations across time. The remaining steps are detailed below.

3.1 Learning the stereo atoms with ASSD

We seek to learn an $L \times L$ dictionary from the signals, $\mathbf{x}_k \in \mathbb{R}^L$, in the columns of the newly formed matrix $\mathbf{X}$, so that $\mathbf{x}_k$ can be represented as [7]

$$\mathbf{x}_k = \sum_{l=1}^{L} \alpha_k^l \boldsymbol{\psi}^l \tag{3}$$

where $\boldsymbol{\psi}^l$ is an atom in the dictionary, $\alpha_k^l, \quad l = 1, \ldots, L$ are the expansion coefficients which encode explicit information regarding the properties of the signal $\mathbf{x}_k$, depending on the choice of dictionary $\mathscr{D}$, and $L \ll n_{\max}$.

The proposed algorithm adaptively learns a data dependent dictionary by sequentially extracting the columns of the matrix $\mathbf{X}$. It is inspired by the idea of setting each new atom equal to the column of $\mathbf{X}$ that satisfies:

$$\max_k \frac{||\mathbf{x}_k||_2}{||\mathbf{x}_k||_1} \tag{4}$$

where $||\cdot||_1$ and $||\cdot||_2$ denote the L1- and L2-norm respectively. Thus at each iteration, the method reduces the energy of the data by a maximum amount, across all frames, while ensuring that the L1-norm is reduced by a minimum amount. In practice, the L1-norm is not re-normalized at each step, and therefore 4 is strictly achieved only for the first atom. The proposed ASSD algorithm solves the maximization problem in equation (4) according to the steps outlined below.

Initialization: At iteration $j = 1$

- ensure that the columns of $\mathbf{X}$ have unit L1-norm

$$\tilde{\mathbf{x}}_k = \frac{\mathbf{x}_k}{||\mathbf{x}_k||_1} \tag{5}$$

  where $\mathbf{x}_k$ the k-th column of $\mathbf{X}$. This leads to a new data matrix $\tilde{\mathbf{X}}$, whose columns now have unit L1-norm. The superscript $\tilde{}$, used hereafter, denotes the normalized matrix and its columns;

- set the residual matrix

$$\mathbf{R}^0 = \tilde{\mathbf{X}} \tag{6}$$

  where $\mathbf{R}^j = [\mathbf{r}_1^j, \ldots, \mathbf{r}_{k_{\max}}^j]$, and $\mathbf{r}_k^j \in \mathbb{R}^{k_{\max}}$ is a residual column vector corresponding to the k-th column of $\mathbf{R}^j$.

Repeat, for all atoms to be extracted:

1. Compute the L2-norm of each frame

$$E_k = ||\mathbf{r}_k^j||_2 = \sum |\mathbf{r}_k^j|^2. \tag{7}$$

2. Find the index $\hat{k}$ corresponding to the signal block with largest L2-norm, $\mathbf{r}_{\hat{k}}^j$

$$\hat{k} = \arg\max_{k \in \mathbb{K}} (E_k) \tag{8}$$

   where $\mathbb{K} = \{1, \ldots, k_{\max}\}$ is the set of all indices pointing to the columns of $\mathbf{R}^j$.

At each iteration $j \in \{1, \ldots, L\}$, the residual vector with highest L2-norm, $\mathbf{r}_{\hat{k}}^j$, becomes a dictionary element, and all residual vectors $\mathbf{r}_k^j$ decrease by an appropriate amount, determined by the selected atom $\boldsymbol{\psi}^j$ and the coefficient of expansion α_k^j.

3. Set the j-th dictionary element $\boldsymbol{\psi}^j$ to be equal to the residual vector with largest L2-norm $\mathbf{r}^j{}_{\hat{k}}$

$$\boldsymbol{\psi}^j = \mathbf{r}_{\hat{k}}^j. \tag{9}$$

4. Evaluate the coefficients of expansion, given by the inner product between the residual vector $\mathbf{r}_k^j$, and the atom $\boldsymbol{\psi}^j$

$$\alpha_k^j = \langle \mathbf{r}_k^j, \boldsymbol{\psi}^j \rangle. \tag{10}$$

5. Compute the new residual, by removing the component along the chosen atom, for each element k in $\mathbf{r}_k^j$

$$\mathbf{r}_k^j = \mathbf{r}_k^{j-1} - \frac{\alpha_k^j}{\langle \boldsymbol{\psi}^j, \boldsymbol{\psi}^j \rangle} \boldsymbol{\psi}^j. \tag{11}$$

The term in the denominator of $\frac{\alpha_k^j}{\langle \boldsymbol{\psi}^j, \boldsymbol{\psi}^j \rangle}$ in equation (11), is included to ensure that the coefficient of expansion α_k^j corresponding to the inner product between the selected atom $\boldsymbol{\psi}^j$ and the frame of maximum L2-norm $\mathbf{r}_{\hat{k}}^j$, is normalized to 1. Then, the corresponding column of the residual matrix $\mathbf{R}^j$ is set to zero, since the whole atom is removed. This step ensures that the transform is orthogonal, hence resulting in a relatively simple source reconstruction step in the SCA procedure, as we shall see in section 3.3.

3.2 Clustering the atom pairs

Having learned a set of L atom pairs $\boldsymbol{\psi}_l^{(i)}$, $l = \{1, \ldots, L\}$, one for each source $i = 1, 2$, we cluster them together into subsets corresponding to each source to be separated, according to their time delay, or direction of arrival (DOA). This is done by finding the time delay τ_l between the atoms in each pair l, using the generalized cross-correlation with phase transform (GCC-PHAT) algorithm [8]

$$R_l(\tau) = \int_{-\infty}^{\infty} \frac{\Psi_l^{(1)}(\omega) \Psi_l^{(2)}(\omega)^*}{|\Psi_l^{(1)}(\omega) \Psi_l^{(2)}(\omega)^*|} e^{j\omega\tau} d\omega \tag{12}$$

where $\Psi_l^{(1)}(\omega), \Psi_l^{(2)}(\omega)$ are the Fourier transforms of the atom pairs $\boldsymbol{\psi}_l^{(1)}$ and $\boldsymbol{\psi}_l^{(2)}$ respectively. The function $R_l(\tau)$ typically exhibits a single sharp peak at the lag corresponding to the time delay between the two signals, which is consistent with the learned atom pairs exhibiting a dominant DOA.

The atoms are subsequently grouped using the K-means clustering algorithm. The time delay 'centroid' T_i, $i = 1,2$ corresponding to each source is found, and we define the index sets

$$\gamma_i = \{l \mid (T_i - \Delta) \leq \tau_l \leq (T_i + \Delta)\} \tag{13}$$

corresponding to the atoms with delays within some threshold Δ of the cluster centroid, reserving a 'discard' cluster

$$\gamma_0 = \{l \mid l \notin \gamma_i, i = 1,2\} \tag{14}$$

for atoms that will not be associated with any of the i sources. Thus, in the space of reshaped vectors $\mathbf{x}_k(n)$, a subspace

$$E_i = \text{span}\{\boldsymbol{\psi}_l^{(i)},\ l \in \gamma_i\},\ i = 1,2 \tag{15}$$

corresponding to each source is defined.

3.3 Reconstructing the sources

To reconstruct the original sources, two mask matrices $\mathbf{H}^{(i)}$, $i = 1,2$ are defined as

$$\mathbf{H}^{(i)} = \text{diag}(h_1^{(i)}, \ldots, h_L^{(i)}) \tag{16}$$

with the diagonal elements of $\mathbf{H}^{(i)}$ set to one or zero depending on whether a transform component is considered to belong to the subspace E_i corresponding to the i-th source. the mask values given by

$$h_l^{(i)} = \begin{cases} 1 & \text{if } l \in \gamma_i \\ 0 & \text{otherwise} \end{cases} \tag{17}$$

for $l = 1, \ldots, L$. Then, the estimated image $\hat{\tilde{\mathbf{X}}}^{(i)}$ of the i-th source is given by

$$\hat{\tilde{\mathbf{X}}}^{(i)}(n) = \mathbf{D}^T \mathbf{H}^{(i)} \mathbf{D} \mathbf{X}^{(i)}. \tag{18}$$

Finally, we use the reverse of the process described in section 3 to find the source image $\hat{\mathbf{x}}^{(i)}(n) = \left[\hat{x}_1^{(i)}(n), \hat{x}_2^{(i)}(n)\right]^T$, that is, the vector of images of the i-th source at both microphones. We refer to this SCA-based source separation algorithm as ASSD-SS.

4. SOURCE SEPARATION WITH K-SVD

The K-SVD algorithm learns an overcomplete dictionary, under the constraint that the signal representation is sparse. K-SVD attempts to minimize the following expression [5]:

$$\min_{\mathbf{D},\mathbf{X}} \{\|\mathbf{Y} - \mathbf{D}\mathbf{X}\|_F^2\} \text{ subject to } \forall i,\ \|\mathbf{x}_i\|_0 \leq T_0 \tag{19}$$

where $\mathbf{Y}$, $\mathbf{X}$ and $\mathbf{D}$ are the signal to be approximated, the coefficient matrix, and the dictionary matrix, respectively; $\|\cdot\|_F$ is the Frobenius norm, and $\|\cdot\|_0$ is the l^0 norm, counting the nonzero entries of a vector. Unlike typical algorithms, the dictionary design and signal decomposition are not conducted separately. Rather, the two steps are performed simultaneously by alternatively fixing the dictionary and finding a signal decomposition, and then updating the dictionary matrix $\mathbf{D}$ one column at the time, while allowing the expansion coefficients to change in this stage [5]. The coefficient update stage can be performed using any approximation pursuit method, as long as the solution has a fixed and predetermined number of nonzero entries, hence imposing a very strong sparsity constraint. In [5], the authors select orthogonal matching pursuit (OMP), as they found the overall algorithm to be more efficient. The dictionary update stage is based on the singular value decomposition of the representation error matrix $\mathbf{Y} - \mathbf{D}\mathbf{X}$, and more precisely, K singular value decomposition computations are performed, each determining a column of the dictionary matrix. A detailed description of the K-SVD algorithm, reproduced from [5], is given as Algorithm 1.

Source separation is performed based on the SCA procedure, as in section 3, but with the dictionary learned with K-SVD replacing that from the ASSD algorithm. The remaining steps are as outlined in section 3. We refer to the resulting source separation algorithm as KSVD-SS.

Algorithm 1 The K-SVD algorithm, reproduced from [5].

Task: Find the best dictionary to represent the data samples $\{\mathbf{y}_i\}_{i=1}^N$ as sparse compositions, by solving

$$\min_{D,X} \{||\mathbf{Y} - \mathbf{D}\mathbf{X}||_F^2\} \text{ subject to } \forall i,\ ||\mathbf{x}_i||_0 \leq T_0$$

Initialization: Set the dictionary matrix $\mathbf{D}^{(0)} \in \mathbb{R}^{n \times K}$ with ℓ^2 normalized columns. Set $J = 1$. Repeat until convergence (stopping rule):

- *Sparse Coding Stage*: Use any pursuit algorithm to compute the representation vectors $\mathbf{x}_i$ for each example $\mathbf{y}_i$, by approximation the solution of

  $$i = 1,2,\ldots,N,\ \min_{\mathbf{x}_i} \{||\mathbf{y}_i - \mathbf{D}\mathbf{x}_i||_2^2\} \text{ subject to } ||\mathbf{x}_i||_0 \leq T_0$$

- *Codebook Update Stage*: For each column $k = 1,2,\ldots,K$ in $\mathbf{D}^{(J-1)}$, update it by
  - Define the group of examples that use this atom, $\omega_k = \{i | 1 \leq i \leq N,\ \mathbf{x}_T^k(i) \neq 0\}$.
  - Compute the overall representation error matrix, $\mathbf{E}_k$, by
    $$\mathbf{E}_k = \mathbf{Y} - \sum_{j \neq k} \mathbf{d}_j \mathbf{x}_T^j.$$
  - Restrict $\mathbf{E}_k$ by choosing only the columns corresponding to ω_k, and obtain $\mathbf{E}_k^R$.
  - Apply SVD decomposition $\mathbf{E}_k^R = \mathbf{U}\boldsymbol{\Delta}\mathbf{V}^T$. Choose the updated dictionary column $\tilde{\mathbf{d}}_k$ to be the first column of $\mathbf{U}$. Update the coefficient vector $\mathbf{x}_R^k$ to be the first column of $\mathbf{V}$ multiplied by $\boldsymbol{\Delta}(1,1)$.
- Set $J = J + 1$.

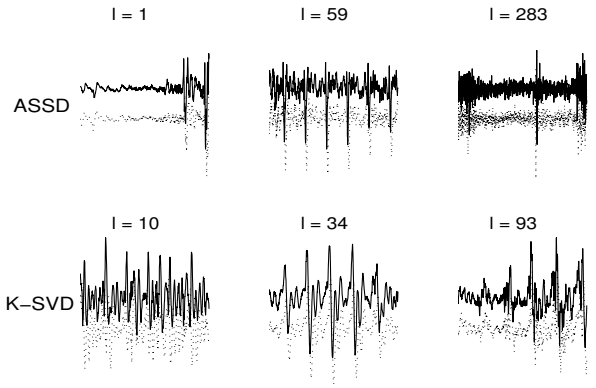

Figure 1: Examples of the atoms pairs learned with the ASSD transform (upper plots) and with the K-SVD algorithm (lower plots). The value l denotes the position of the atom within the dictionary.

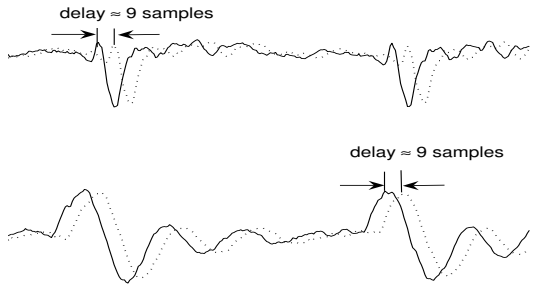

Figure 2: Example of an atom pair learned with the ASSD algorithm (upper plot) and with the K-SVD method (lower plot).

4.1 Computational complexity

Since the two separation methods differ only in the dictionary learning algorithm, it is sufficient to compare the complexity of ASSD and K-SVD. The most computationally expensive step in the ASSD algorithm is the evaluation of the coefficients of expansion in equation (10). Evaluation of α_k^j, for a residual $\mathbf{r}_k^j$ of length $k_{\max}$ and an atom $\boldsymbol{\psi}^j$ of length L, is $\mathrm{O}(Lk_{\max})$ for each iteration $j \in \{1,\ldots,L\}$, thus resulting in an overall complexity of $\mathrm{O}(L^2 k_{\max})$. The K-SVD algorithm entails performing K (or $k_{\max}$ in the case discussed here) computationally intensive singular value decomposition steps. In general, the complexity of the SVD transform is $\mathrm{O}(Lk_{\max}^2)$, and this had to be applied $k_{\max}$ times, each corresponding to a column of the dictionary matrix, thus giving a computational complexity of $\mathrm{O}(Lk_{\max}^3)$. Note that, since the atoms are extracted from the columns of $\mathbf{R}^j$, their number is at most equal to $k_{\max}$, i.e. $L \leq k_{\max}$, and therefore $\mathrm{O}(Lk_{\max}^2) < \mathrm{O}(Lk_{\max}^3)$. In the next section, we will see what these computational times correspond to in real time when performing computer simulations.

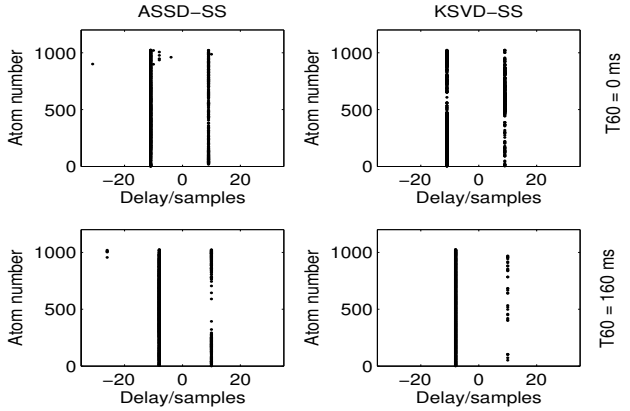

Figure 3: Plot of the time delays estimated for ASSD-SS and KSVD-SS, under anechoic (upper plot), and echoic (lower plot) mixing.

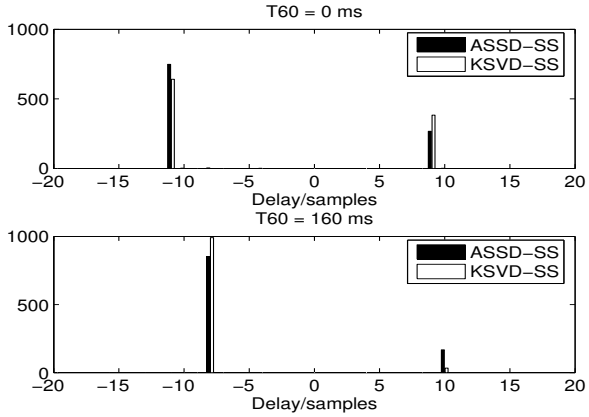

Figure 4: Plot of the histogram of the time delays estimated for ASSD-SS and KSVD-SS, under anechoic (upper plot), and echoic (lower plot) mixing.

5. EXPERIMENTAL RESULTS

Some of the atoms obtained with the proposed ASSD algorithm, when two male speech signals were synthetically mixed in the presence of only time delays (anechoic mixing), and with delays of 9 and -11 samples, are shown in the upper part of figure 1. The sources and sensors had the same layout as in [6]. They were obtained using a frame length of $N = 1024$ samples for ASSD, and an overlap of $T = 1014$ samples. Examples of the atoms generated by K-SVD, using the parameters selected in [5] are shown in the lower part of figure 1. In both cases, the atoms capture characteristics of the speech signals. In particular, many of the atoms learned with ASSD were found to be quite localized. Figure 2 shows a zoomed in view of a typical example of an atom pair of length 1024 samples, learned with the ASSD (upper plot) and K-SVD (lower plot) algorithms, from the same stereo mixture described above. The plots illustrate that the learned atom pairs encode information about the time-delay and amplitude differences in the mixing channel. This suggests that each atom pair relates to a particular source.

Reverberation	Method	SAR	SIR	SDR
0 msec	ASSD-SS	2.8	14.1	2.2
	KSVD-SS	1.3	13.8	0.8
160 msec	ASSD-SS	2.6	14.4	2.2
	KSVD-SS	3.5	15.5	3.0

Table 1: Objective performance of ASSD-SS and KSVD-SS. All values are expressed in decibels (dB).

Next, the two methods proposed here were used to separate the sources in the anechoic mixture case, and when reverberation of 160 ms was present. The position of the sources and sensors was such that the direction of arrival of the source signals corresponded to delays of 9 and -11 samples in the anechoic case, and 10 and -8 in the case of convolutive mixtures. The performance of the two algorithms was evaluated using the objective criteria of Signal-to-Distortion Ratio (SDR), Signal-to-Interference Ratio (SIR) and Signal-to-Artefacts Ratio (SAR), as defined in [9]. The SDR ratio measures the difference between an estimated source and a target source, allowing for possible linear filtering between them; for this reason, we allowed for time-invariant filtering of filter length 1024 samples when calculating SDR. SIR and SAR measure, respectively, the distortion due to interfering sources and other artefacts. Table 1 shows the values obtained for the two methods, and the single figures for all sources were obtained by averaging the criteria across all microphones and sources. Figure 3 shows the time-delays estimated with the GCC-PHAT algorithm, for the ASSD-SS and KSVD-SS methods, for the anechoic and convolutive mixing cases, while in figure 4 the histograms of the estimated time-delays are compared.

It can be seen that in the anechoic case both methods correctly identify the direction of arrival of the two sources. The objective measures indicate that ASSD-SS gives better separation results overall, with low interference from the other source, and fewer artefacts. It is also interesting to note how the behavior of KSVD-SS in particular changes in the presence of reverberations, when fewer atoms are assigned to the source with a delay of 10 samples. However, the histogram in figure 4 shows that most atoms are assigned to the other source. The objective measures in Table 1 seem to suggest that this is an advantage, with KSVD-SS outperforming ASSD-SS in this case. Nonetheless, if we look at the objective measures for each source, averaged over the sensors, shown in table 2, we find that for the source from direction -8 samples (source 2 in the figure), KSVD-SS performs better than in the anechoic case, while for the other source, performance is poor.
Finally, the speed of learning of KSVD-SS was found to be quite slow, as discussed in section 4.1. In our simulations for the experiments described above, conducted on a Pentium IV at 3.4GHz, using Matlab Version 7.0.4 (R14SP2), and under the Microsoft Windows XP operating system, the computation time for ASSD-SS was about 17 minutes (1036 sec), while the KSVD-SS algorithm required more than 3 hours (12650 sec), that is, separation with ASSD-SS was over 10 times faster than with KSVD-SS.

Reverb.	Method	Source	SAR	SIR	SDR
160 msec	KSVD-SS	source 1	1.9	11.4	1.2
		source 2	5.2	19.6	4.9

Table 2: Objective performance of KSVD-SS for the two sources, averaged across the sensors.

6. CONCLUSIONS

We have presented a source separation algorithm that addresses the problem for echoic and anechoic environments, based on a sparse component analysis type approach. The ASSD-SS method uses a novel adaptive stereo sparse dictionary learning algorithm that finds atom pairs simultaneously from the two channels. A similar method using K-SVD for the dictionary learning step was also considered, and it was found that ASSD-SS is faster than the latter. Both algorithms were found to correctly identify the direction of arrival of the sources, and to separate them both in anechoic and echoic mixing conditions.

REFERENCES

[1] R. Gribonval, "Sparse decomposition of stereo signals with matching pursuit and application to blind separation of more than two sources from a stereo mixture," in *Proc. of ICASSP*, 2002, vol. 3, pp. 3057–3060.

[2] R. Gribonval and S. Lesage, "A survey of sparse component analysis for blind source separation: principles, perspectives, and new challanges," in *Proc. of ESANN*, 2006, pp. 323–330.

[3] Ö. Yilmaz and S. Rickard, "Blind separation of speech mixtures via time-frequency masking," *IEEE Trans. on Signal Processing*, vol. 52, pp. 1830–1847, 2004.

[4] S. A. Abdallah and M. D. Plumbley, "Application of geometric dependency analysis to the separation of convolved mixtures," in *Proc. of ICA*, 2004, pp. 22–24.

[5] Michal Aharon, Michael Elad, and Alfred Bruckstein, "K-SVD: An algorithm for designing overcomplete dictionaries for sparse representations," *IEEE Trans. on Signal Processing*, vol. 54, pp. 4311–4322, 2006.

[6] M. G. Jafari, E. Vincent, S. A. Abdallah, M. D. Plumbley, and M. E. Davies, "An adaptive stereo basis method for convolutive blind audio source separation," *Neurocomputing*, 2008. To appear.

[7] M. Goodwin and M. Vetterli, "Matching pursuit and atomic signal models based on recursive filter banks," *IEEE Trans. on Signal Processing*, vol. 47, pp. 1890–1902, 1999.

[8] C. Knapp and G. Carter, "The generalized correlation method for estimation of time delay," *IEEE Trans. on Acoustic, Speech, and Signal Processing*, vol. 24, pp. 320–327, 1976.

[9] C. Févotte, R. Gribonval, and E. Vincent, "BSS_EVAL toolbox user guide," Tech. Rep. 1706, IRISA, http://www.irisa.fr/metiss/bss_eval/, 2005.