

ADAPTIVE FUSION OF DICTIONARY LEARNING AND MULTICHANNEL BSS

Vahid Abolghasemi^{1,2}, Saideh Ferdowsi¹, Bahador Makkiabadi^{1,3}, and Saeid Sanei¹

¹Faculty of Engineering and Physical Sciences, University of Surrey, Guildford, UK

²School of Engineering and Design, Brunel University, Uxbridge, UK

³Electrical Engineering Department, Islamic Azad University, Ashtian Branch, Ashtian, IRAN
vabolghasemi@ieee.org, {s.ferdowsi, b.makkiabadi, s.sanei}@surrey.ac.uk

ABSTRACT

Sparsity has been shown to be very useful in blind source separation. However, in most cases the sources of interest are not sparse in their current domain and are traditionally sparsified using a predefined transform or a learned dictionary. In this paper, we address the case where the underlying sparse domains of the sources are not available and propose a solution via fusing the dictionary learning into the source separation. In the proposed method, a local dictionary is learned for each source along with separation and denoising of the sources. This iterative procedure adapts the dictionaries to the corresponding sources which consequently improves the quality of source separation. The results of our experiments are promising and confirm the strength of the proposed approach.

Index Terms— Blind source separation, dictionary learning, image denoising, morphological component analysis, sparsity.

1. INTRODUCTION

The instantaneous blind source separation (BSS) problem can be presented by the following linear mixing model:

$$\mathbf{Y} = \mathbf{A}\mathbf{X} + \mathbf{V}, \quad (1)$$

where $\mathbf{Y} \in \mathbb{R}^{m \times N}$, composed of m row vectors $\{\mathbf{y}^j\}_{j=1}^m$, is the observation matrix, $\mathbf{X} \in \mathbb{R}^{n \times N}$, composed of n row vectors $\{\mathbf{x}^j\}_{j=1}^n$, is the source matrix and \mathbf{A} is the $m \times n$ mixing matrix. The additive noise matrix of size $m \times N$ is denoted by \mathbf{V} . In BSS the aim is to estimate both \mathbf{A} and \mathbf{X} from the observations. Exploiting statistical *independency* of the sources, using independent component analysis (ICA) techniques [1], is one of the popular approaches for solving (1). Another useful assumption is *sparsity* of the sources which has been received much attention recently. The term *sparse* refers to signals/images with small number of non-zero samples. In sparse component analysis (SCA) it has been shown that sparsity improves the source diversity and enables the separation of the sources [2]. In a recent advanced technique, called multichannel morphological component analysis (MMCA) [3, 4], the main assumption is that each source can be sparsely represented in a specific known transform domain. It is an extension of previously proposed method of morphological component analysis (MCA) [5–8] to the multichannel case. In MCA the given signal/image is decomposed into different morphological components subject to sparsity of each component in a known basis (or dictionary). MMCA performs well when *a priori* knowledge about the sparsifying transforms for each individual source is available; but what if such *a priori* is not available or what if the sources are not sparsely representable using the predefined basis?

One recent approach has been proposed in [9] which attempts to learn a single dictionary from the mixture of two images and then applies a decision criterion to the dictionary atoms to separate the images. In another recent work, Peyre *et al.* [10] presented an adaptive MCA scheme by learning the morphologies of image layers. They proposed to use both adaptive *local* dictionaries and fixed *global* transforms (e.g. wavelet, curvelet) for image separation from a single mixture. However, both the above methods are limited to the case of single mixture of two additive images plus noise.

In this paper we consider the general BSS model (with $m \geq n$) where the sparsifying dictionaries/transforms are not available. We propose a novel approach to adaptively learn the dictionaries from the mixed images within the source separation process. This method is motivated by the idea of image denoising using a learned dictionary from the corrupted image in [11]. We start by theoretically extending the denoising problem to BSS. Then, a practical algorithm is proposed for BSS without any prior knowledge about the sparsifying basis. The results indicate that the adaptive dictionary learning, separately for each source, enhances the separability of the sources.

In the next section the problem formulation followed by theoretical extension of image denoising to source separation is given. In section 3 a practical approach for this purpose is proposed. The numerical results are given in section 4. Finally, the paper is concluded in section 5.

2. PROBLEM FORMULATION

2.1. Image denoising

Consider Gaussian noise added to an image, i.e. $\mathcal{Y} = \mathcal{X} + \mathcal{V}$. Elad *et al.* [11] proposed an image denoising method via learning a dictionary using the corrupted image itself. They use small (overlapped) patches of noisy image, \mathcal{Y} , to learn a sparsifying dictionary using K-SVD (see [12]). The obtained dictionary is called *local*, since it describes the local features extracted from small patches. Let us represent the noisy $\sqrt{N} \times \sqrt{N}$ image \mathcal{Y} by vector \mathbf{y} of length N . The unknown denoised image \mathcal{X} is also vectorized and represented as \mathbf{x} . The i -th $\sqrt{r} \times \sqrt{r}$ patch from \mathcal{X} is shown by vector $\mathfrak{R}_i \mathbf{x}$ which is of length $r \ll N$. For notational simplicity the i -th patch is expressed as explicit multiplication of operator \mathfrak{R}_i (as a binary $r \times N$ matrix) by \mathbf{x} . However, in practice, we apply a nonlinear operation to extract the patches from image \mathcal{X} by sliding a mask of appropriate size over the entire image. Given that, the overall denoising problem in [11] is expressed as

$$\min_{\mathbf{D}, \{\mathbf{s}_i\}, \mathbf{x}} \lambda \|\mathbf{y} - \mathbf{x}\|_2^2 + \sum_{i=1}^p [\mu_i \|\mathbf{s}_i\|_0 + \|\mathbf{D}\mathbf{s}_i - \mathfrak{R}_i \mathbf{x}\|_2^2], \quad (2)$$

where scalars λ and μ control the noise power and sparsity degree, respectively. Also, $\mathbf{D} \in \mathbb{R}^{r \times k}$ is the sparsifying dictionary which contains normalized columns (also called atoms) and $\{\mathbf{s}_i\}$ are sparse coefficients of length k . $\|\cdot\|_0$ denotes the ℓ_0 -norm and counts the number of non-zeros. We note that index i , throughout the paper, indicates the i -th patch. Furthermore, the total number of extracted patches from an image is denoted by p .

In [11], the minimization of (2) starts with extracting and rearranging all the patches of \mathcal{X} . The patches are then processed by K-SVD [12] which updates \mathbf{D} and estimates the sparse coefficients $\{\mathbf{s}_i\}_{i=1}^p$. Afterward, \mathbf{D} and $\{\mathbf{s}_i\}_{i=1}^p$ are assumed fixed and \mathbf{x} is estimated by computing

$$\hat{\mathbf{x}} = (\lambda \mathbf{I} + \sum_{i=1}^p \mathfrak{R}_i^T \mathfrak{R}_i)^{-1} (\lambda \mathbf{y} + \sum_{i=1}^p \mathfrak{R}_i^T \mathbf{D} \mathbf{s}_i), \quad (3)$$

where \mathbf{I} is the identity matrix and $(\cdot)^T$ denotes transpose operation. Again, \mathbf{D} and $\{\mathbf{s}_i\}$ are updated by K-SVD but this time using the

patches from $\hat{\mathbf{x}}$ which are less noisy. Such conjoined denoising and dictionary adaptation is repeated to minimize (2). In practice, (3) is obtained computationally easier since $\sum_i \mathfrak{R}_i^T \mathfrak{R}_i$ is diagonal and (3) can be calculated in a pixel-wise fashion. It is seen that (3) is a kind of averaging using both noisy and denoised patches, which if repeated along with updating of other parameters, will denoise the entire image [11].

2.2. Multichannel source separation

In this section the aim is to extend the denoising problem (2) to the multichannel source separation. Consider the BSS model introduced in section 1, and assume further that the sources of interest are 2-D grayscale images. The BSS model for 2-D sources can be represented by vectorizing all images $\{\mathcal{X}_1, \dots, \mathcal{X}_n\}$ to $\{\mathbf{x}_1, \dots, \mathbf{x}_n\}$ and then stacking them to form $\mathbf{X} = [\mathbf{x}_1^T, \dots, \mathbf{x}_n^T]^T$. The BSS model (1) cannot be directly incorporated into (2) as we require both \mathbf{X} and \mathbf{Y} to be single vectors. Hence, we use the vectorized versions of these matrices in model (1) which can be obtained using the properties of the Kronecker product¹:

$$\mathbf{y} = (\mathbf{I} \otimes \mathbf{A}) \mathbf{x} + \mathbf{v}. \quad (4)$$

In the above expression $\mathbf{x} = \text{vec}(\mathbf{X})$ and $\mathbf{y} = \text{vec}(\mathbf{Y})$ are column vectors of length nN and mN , respectively. $(\mathbf{I} \otimes \mathbf{A})$ is a block diagonal matrix of size $mN \times nN$, and \otimes is the Kronecker product symbol. We consider the noiseless setting and modify (2) to

$$\min_{\mathbf{D}, \{\mathbf{s}_i\}, \mathbf{x}} \lambda \|\mathbf{y} - (\mathbf{I} \otimes \mathbf{A}) \mathbf{x}\|_2^2 + \sum_{i=1}^p [\mu_i \|\mathbf{s}_i\|_0 + \|\mathbf{D} \mathbf{s}_i - \mathfrak{R}_i \mathbf{x}\|_2^2]. \quad (5)$$

It is seen that the above expression is similar to (2) with an extra variable (i.e. mixing matrix \mathbf{A}) which should be estimated. In addition, the vectors \mathbf{x} and \mathbf{y} are much lengthier than \mathbf{x} and \mathbf{y} as they represent vectorized versions of multiple images and mixtures. The same principle as that explained in section 2.1 can be applied to update \mathbf{D} and $\{\mathbf{s}_i\}$. However, estimation of \mathbf{x} is slightly different from that of \mathbf{x} in (3). It has a closed form solution which is obtained by taking the gradient of (5) and setting it to zero:

$$0 = \lambda (\mathbf{I} \otimes \mathbf{A})^T ((\mathbf{I} \otimes \mathbf{A}) \mathbf{x} - \mathbf{y}) + \sum_{i=1}^p \mathfrak{R}_i^T (\mathfrak{R}_i \mathbf{x} - \mathbf{D} \mathbf{s}_i), \quad (6)$$

which leads to:

$$\begin{aligned} \hat{\mathbf{x}} &= (\lambda (\mathbf{I} \otimes \mathbf{A})^T (\mathbf{I} \otimes \mathbf{A}) + \sum_{i=1}^p \mathfrak{R}_i^T \mathfrak{R}_i)^{-1} \dots \\ &\dots (\lambda (\mathbf{I} \otimes \mathbf{A})^T \mathbf{y} + \sum_{i=1}^p \mathfrak{R}_i^T \mathbf{D} \mathbf{s}_i). \end{aligned} \quad (7)$$

In order to estimate \mathbf{A} we consider all unknowns, except \mathbf{A} , fixed and simplify (5) by converting the first quadratic term into ordinary matrix product and obtain:

$$\min_{\mathbf{A}} \lambda \|\mathbf{Y} - \mathbf{A} \mathbf{X}\|_F^2. \quad (8)$$

The above minimization problem is easily solved using the pseudo-inverse of \mathbf{X} as:

$$\hat{\mathbf{A}} = \mathbf{Y} \mathbf{X}^T (\mathbf{X} \mathbf{X}^T)^{-1}. \quad (9)$$

The above steps (i.e. estimation of \mathbf{D} , $\{\mathbf{s}_i\}$, \mathbf{x} and \mathbf{A}) should be alternately repeated to minimize (5). However, the long expression (7) is not practically computable, especially if the numbers of sources and observations are large. That is because of dealing with huge $mN \times nN$ matrix $(\mathbf{I} \otimes \mathbf{A})$. In addition, in contrast to the aforementioned denoising problem, the matrix to be inverted in (7) is not diagonal and so, the estimation of \mathbf{x} cannot be calculated pixel-by-pixel, which makes the situation more difficult to handle. In the next section a practical approach is proposed to solve this problem.

¹The matrix multiplication can be expressed as a linear transformation on matrices. In particular: $\text{vec}(\mathbf{ABC}) = (\mathbf{C}^T \otimes \mathbf{A}) \text{vec}(\mathbf{B})$ for three matrices \mathbf{A} , \mathbf{B} and \mathbf{C} . In our case, $\text{vec}(\mathbf{AX}) = (\mathbf{I} \otimes \mathbf{A}) \text{vec}(\mathbf{X})$.

3. ALGORITHM

In order to find a practical solution for (5) we use a hierarchical scheme such as the one in MMCA [3]. To do this, the BSS model of (1) is broken into n rank-1 multiplications and the following minimization problem is defined:

$$\min_{\{\mathbf{a}_j, \mathbf{D}_j, \{\mathbf{s}_i\}, \mathbf{x}^j\}} \lambda_j \|\mathbf{E}_j - \mathbf{a}_j \mathbf{x}^j\|_F^2 + \sum_{i=1}^p [\mu_i \|\mathbf{s}_i\|_0 + \|\mathbf{D}_j \mathbf{s}_i - \mathfrak{R}_i \mathbf{x}^j\|_2^2], \quad (10)$$

where $\|\cdot\|_F$ stands for the Frobenius norm, \mathbf{D}_j denotes the dictionary corresponding to j -th source, i.e. \mathbf{x}^j , and \mathbf{a}_j indicates the j -th column of \mathbf{A} . It is worth mentioning that the patch extractor, \mathfrak{R} , operates similarly on all the image sources $\{\mathbf{x}^j\}_{j=1}^n$ and thus we do not include the source index j for it in our formulations. The j -th residual \mathbf{E}_j in (10) is expressed as:

$$\mathbf{E}_j = \mathbf{Y} - \sum_{\substack{l=1 \\ l \neq j}}^n \mathbf{a}_l \mathbf{x}^l. \quad (11)$$

The main advantage of formulations (10) and (11) is a significant reduction in dimensions (compared with (5)) and the possibility to calculate the image sources pixel-wise. In addition, learning an individual dictionary for each source within the separation process increases the diversity of the sources adaptively.

The solution for (10) is achieved using an alternating scheme. The minimization process for j -th hierarchy can be expressed as follows. The image patches are extracted from j -th image source, \mathcal{X}_j (equivalent to \mathbf{x}^j), and then processed by K-SVD for learning \mathbf{D}_j and estimating sparse coefficients $\{\mathbf{s}_i\}_{i=1}^p$, while other parameters are kept fixed. Then, the gradient of (10) with respect to \mathbf{x}^j is calculated and set to zero:

$$\begin{aligned} 0 &= \lambda_j \mathbf{a}_j^T (\mathbf{a}_j \mathbf{x}^j - \mathbf{E}_j) + \sum_{i=1}^p (\mathbf{x}^j \mathfrak{R}_i^T - \mathbf{s}_i^T \mathbf{D}_j^T) \mathfrak{R}_i = \\ &\lambda_j \mathbf{x}^j + \mathbf{x}^j \sum_{i=1}^p \mathfrak{R}_i^T \mathfrak{R}_i - \lambda_j \mathbf{a}_j^T \mathbf{E}_j - \sum_{i=1}^p \mathbf{s}_i^T \mathbf{D}_j^T \mathfrak{R}_i. \end{aligned} \quad (12)$$

Finally:

$$\hat{\mathbf{x}}^j = (\lambda_j \mathbf{a}_j^T \mathbf{E}_j + \sum_{i=1}^p \mathbf{s}_i^T \mathbf{D}_j^T \mathfrak{R}_i) (\lambda_j \mathbf{I} + \sum_{i=1}^p \mathfrak{R}_i^T \mathfrak{R}_i)^{-1}. \quad (13)$$

It is interesting to notice that the inverting term in the above expression is the same as that in (3) for the denoising problem. Thus, as mentioned before, this calculation can be obtained pixel-wise.

Next, in order to update \mathbf{a}_j a simple least square linear regression such as the one in [3] will give the solution:

$$\hat{\mathbf{a}}_j = \mathbf{E}_j \mathbf{x}^j{}^T. \quad (14)$$

However, a normalization step for $\hat{\mathbf{a}}_j$ is necessary after each update to preserve the column norm of the mixing matrix. The above steps for updating all variables are executed for all j from 1 to n . Moreover, the entire procedure should be repeated to minimize (10). A pseudo-code of the proposed algorithm is given in Algorithm 1.

As already implied, the motivation behind the proposed algorithm is to learn source-specific dictionaries offering sparse representations. Indeed, in the first few iterations of Algorithm 1 each source includes portions of other sources. However, the dictionaries gradually learn the dominant components and reject the weak portions caused by other sources. Using these dictionaries, the estimated sources—which are used for dictionary learning in the next iteration—will have less amount of irrelevant components. This adaptive process is repeated until most of the irrelevant portions are rejected and the dictionaries become source-specific. All the above procedure is carried out to minimize the cost function in (10).

Similar to the denoising method in [11], the proposed method also requires the knowledge about noise power. This should be utilized in the sparse coding step of the K-SVD algorithm for solving the following problem:

Algorithm 1: The proposed algorithm.

Input: Observation matrix \mathbf{Y} , patch size r , dictionary size k , number of sources n , noise standard deviation σ_v , regularization parameter λ , and number of iterations M .

Output: Dictionaries $\{\mathbf{D}_j\}_{j=1}^n$, sparse coefficients $\{\mathbf{s}_i\}$, sources matrix \mathbf{X} , and mixing matrix \mathbf{A} .

- 1 Initialize all \mathbf{D}_j 's with discrete cosine transform (DCT) bases.
- 2 Initialize \mathbf{A} to a random matrix, and set $\mathbf{X} \leftarrow \mathbf{A}^T \mathbf{Y}$.
- 3 Choose σ to be multiple times of σ_v , and set $\Delta\sigma = \frac{1}{M}(\sigma - \sigma_v)$.
- 4 **repeat**
- 5 $\lambda = 30/\sigma$;
- 6 **for** $j=1$ **to** n **do**
- 7 Extract all the patches from \mathbf{x}^j ;
- 8 Apply K-SVD to the patches and update $\{\mathbf{s}_i\}_{i=1}^p$ and \mathbf{D}_j ;
- 9 Calculate the residual using (11);
- 10 Compute \mathbf{x}^j using (13);
- 11 $\mathbf{a}_j \leftarrow \mathbf{E}_j \mathbf{x}^{jT}$;
- 12 $\mathbf{a}_j \leftarrow \frac{1}{\|\mathbf{a}_j\|_2} \mathbf{a}_j$;
- 13 **end**
- 14 $\sigma \leftarrow \sigma - \Delta\sigma$;
- 15 **until** stopping criterion is met;

$$\forall i : \min_{\mathbf{s}_i} \|\mathbf{s}_i\|_0 \text{ s.t. } \|\mathbf{D}_j \mathbf{s}_i - \mathbf{R}_i \mathbf{x}^{jT}\|_2^2 \leq (C\sigma_v)^2, \quad (15)$$

where C is a constant and σ_v is the noise standard deviation. Incorporating the noise power in solving (15) has an important advantage in the dictionary update stage. It ensures that the dictionary does not learn the existing noise in the patches. Consequently, the estimated image using this dictionary would become cleaner which will later refine the dictionary atoms in the next iteration. This progressive denoising loop is repeated until a clean image is achieved.

In the proposed method, however, the *noise* means the portions of other sources which are mixed with \mathbf{x}^j . Hence, we initially consider a high value for $(C\sigma_v)^2$, since the portions of other sources might be high even though the noise is zero, and then gradually reduce it to zero as the iterations evolve. Nevertheless, if the observations themselves are noisy, then, we should start from a higher bound and decrease it toward the actual noise power as the iterations evolve. This annealing scheme is shown in Algorithm 1.

4. RESULTS

In the first experiment we illustrate the results of applying Algorithm 1 to the mixtures of four sources. A 6×4 full-rank random column-normalized \mathbf{A} was used as the mixing matrix. 500 iterations was selected as the stopping criterion. The mixtures were noiseless. However, we selected $\lambda = 30/\sigma_v$ (below see the reason for this choice) and used a decreasing σ_v starting from 10 and reaching to 0.01 at the end of iterations. The patches had 50% overlap. Other parameters were chosen as follows: $r = 64$, $k = 256$, $N = 128 \times 128$, and $C = 1$. Figure 1 illustrates the results of the proposed method together with the results of two ICA-based methods, namely FastICA² and SMICA (spectral matching ICA) [13]. It is seen from this figure that the proposed method could recover the image sources with significantly smaller MSEs (mean square errors) among other methods, while it does not use any prior knowledge about sparsifying basis. In addition, the obtained dictionaries, shown at the bottom row of Figure 1, are well adapted to the corresponding sources. These results confirm the effectiveness of the proposed method over the ICA-based approaches for image source separation.

In the next experiment, the performance of the proposed method in noisy situations is evaluated. For this purpose, we generated four mixtures from two images: *Lena* and *Boat* (Figure 2 (a)). Then, Gaussian noise with standard deviation of 15 was added to the mixture so that the PSNR of the mixture equaled 10 dB. The algorithm started with $\sigma_v = 25$ and evolved while σ_v

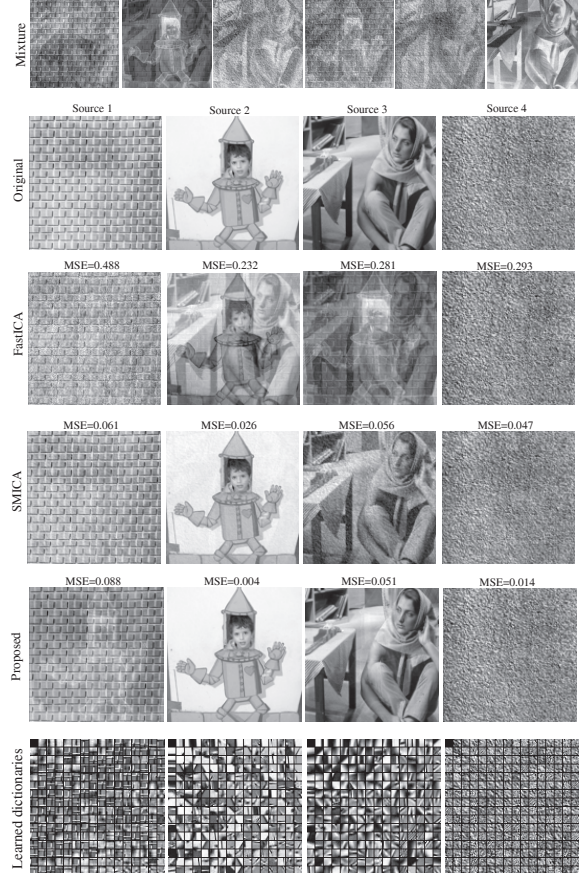


Fig. 1: Separation results of six noiseless mixtures.

was gradually decreasing to $\sigma_v = 15$. We used fully overlapped patches and 200 iterations, with the rest of the parameters similar to the previous experiment. One of the considerable advantages of the proposed method is the ability to denoise the sources during the separation. This is because the core idea of the proposed method comes from a denoising task. In contrast, in most conventional BSS methods the denoising should be carried out either before or after the separation, which is not ideal and may lead the algorithm to fail. The results of this experiment are demonstrated in Figure 2. It is seen from Figure 2 that both separation and denoising tasks have been successful. The corresponding high PSNRs shown in Figure 2 confirm this observation.

Next, in order to find an optimal λ , we investigated the effects of choosing different λ s on the recovery quality. As expected and also shown in [11], selection of λ is dependent on the noise standard deviation, i.e. σ_v . However, our case is slightly different due to the source separation task. We empirically observed better performance by starting with a higher value of σ_v for solving (15) and decreasing it toward the true noise standard deviation, as the iterations proceed. In our simulations Gaussian noises with different power were added to the mixtures and the *mixing matrix criterion*— $\Gamma_{\mathbf{A}} = \|\mathbf{I} - \mathbf{P}\mathbf{A}^{-1}\mathbf{A}\|_1$, where \mathbf{P} is the scaling and permutation matrix— was calculated while varying λ . Figure 3 represents the achieved results for this experiment. From this figure, it can be found that all the curves achieve nearly the same recovery quality for $\lambda\sigma_v = 30 \sim 40$. Therefore, $\lambda \approx 30/\sigma_v$ is a reasonable choice. In the noiseless case where $\sigma_v = 0$ a high value (normally 100) is chosen for λ . Interestingly, these results and the range of appropriate choices for λ are very similar to what achieved in [11].

It is expected that the computation time of the proposed method increases for large number of sources. A set of simulations were

²Available at <http://research.ics.tkk.fi/ica/fastica/>

Table 1: The detailed simulation time of Algorithm 1. “Total” means the total time (in second) elapsed per one iteration.

| Image size: $\sqrt{N} \times \sqrt{N}$ | 128 × 128 | | | | | | 256 × 256 | | | | | |
|---|-----------|-------|-------|-------|-------|-------|-----------|-------|-------|-------|-------|--------------|
| Mixing matrix size: m, n | 4, 2 | 4, 2 | 8, 4 | 8, 4 | 16, 8 | 16, 8 | 4, 2 | 4, 2 | 8, 4 | 8, 4 | 16, 8 | 16, 8 |
| Patch size: r | 16 | 64 | 16 | 64 | 16 | 64 | 16 | 64 | 16 | 64 | 16 | 64 |
| Dictionary atom size: k | 64 | 256 | 64 | 256 | 64 | 256 | 64 | 256 | 64 | 256 | 64 | 256 |
| Computation time (sec): | | | | | | | | | | | | |
| Sparse coding (Batch-OMP [14]) | 0.032 | 0.059 | 0.073 | 0.133 | 0.134 | 0.172 | 0.116 | 0.184 | 0.354 | 0.747 | 0.760 | 0.965 |
| Dictionary update | 0.366 | 1.229 | 0.724 | 2.314 | 1.450 | 4.391 | 0.565 | 1.594 | 1.532 | 4.786 | 2.772 | 7.675 |
| Residual calculation | 0.001 | 0.002 | 0.003 | 0.003 | 0.007 | 0.007 | 0.008 | 0.008 | 0.016 | 0.016 | 0.038 | 0.038 |
| \hat{x}^j estimation | 0.081 | 0.023 | 0.161 | 0.047 | 0.324 | 0.094 | 0.325 | 0.93 | 0.654 | 0.196 | 1.326 | 0.395 |
| \mathbf{a}_j update and normalization | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.001 | 0.001 | 0.001 | 0.002 | 0.002 | 0.004 | 0.004 |
| Total | 0.540 | 1.333 | 1.091 | 2.547 | 2.229 | 4.815 | 1.260 | 1.961 | 3.104 | 5.966 | 6.300 | 9.802 |

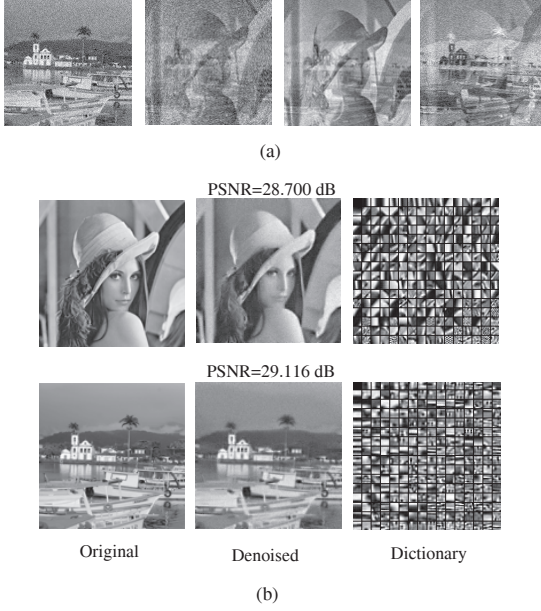


Fig. 2: (a) Noisy mixtures, (b) separation results.

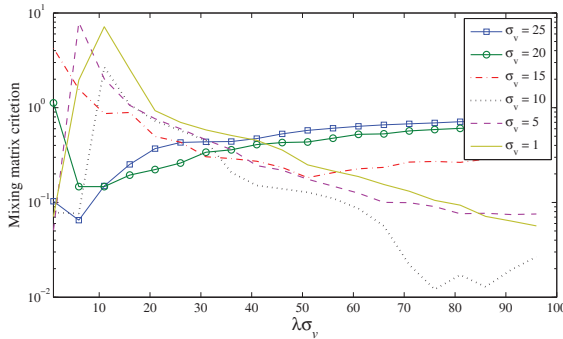


Fig. 3: Mixing matrix criterion as a function of $\lambda\sigma_v$.

conducted to numerically demonstrate this effect and also to evaluate the effects of changing other parameters such as patch size and dictionary size on the simulation time. In Table 1 we have given these results per one iteration of Algorithm 1, where the image patches had 50% overlap. A desktop computer with a Core 2 Due CPU of 3 GHz and 2 GB of RAM was used for this experiment. It is interesting to note that as seen from Table 1 much of the computational burden is incurred by the dictionary update stage (using K-SVD). In addition, the computation of \hat{x}^j is far less complicated than both “sparse coding” and “dictionary update”, due to pixel-wise operation. This implies that further effort is required to speed up the dictionary learning part of the proposed algorithm.

5. CONCLUSION

In this paper the multichannel source separation problem has been addressed. Unlike the existing sparsity-based methods, we assumed there is no prior knowledge about the underlying sparsity domain of the sources. Instead, we have proposed to fuse learning adaptive dictionaries for each individual source into the separation process. Our simulation results on both noisy and noiseless mixtures have been encouraging and confirmed the effectiveness of the proposed approach.

6. REFERENCES

- [1] A. Hyvärinen, J. Karhunen, and E. Oja, *Independent Component Analysis*, Wiley-Interscience, May 2001.
- [2] M. Zibulevsky and B. A. Pearlmutter, “Blind source separation by sparse decomposition in a signal dictionary,” *Neural Comput.*, vol. 13, pp. 863–882, April 2001.
- [3] J. Bobin, Y. Moudden, J. Starck, and M. Elad, “Morphological diversity and source separation,” *IEEE Signal Processing Letters*, vol. 13, no. 7, pp. 409–412, July 2006.
- [4] J. Bobin, J. Starck, J. Fadili, and Y. Moudden, “Sparsity and morphological diversity in blind source separation,” *IEEE Trans. on Image Processing*, vol. 16, no. 11, pp. 2662–2674, November 2007.
- [5] J. Starck, M. Elad, and D. Donoho, “Redundant multiscale transforms and their application for morphological component separation,” *Advances in Imaging and Electron Physics*, vol. 132, pp. 287–348, 2004.
- [6] J. Starck, M. Elad, and D. Donoho, “Image decomposition via the combination of sparse representations and a variational approach,” *IEEE Trans. on Image Processing*, vol. 14, no. 10, pp. 1570–1582, October 2005.
- [7] M. Elad, J. Starck, P. Querre, and D. Donoho, “Simultaneous cartoon and texture image inpainting using morphological component analysis (MCA),” *Applied and Computational Harmonic Analysis*, vol. 19, no. 3, pp. 340–358, November 2005.
- [8] J. Bobin, J. Starck, J. Fadili, Y. Moudden, and D. Donoho, “Morphological component analysis: An adaptive thresholding strategy,” *IEEE Trans. on Image Processing*, vol. 16, no. 11, pp. 2675–2681, November 2007.
- [9] M. Elad, *Sparse and Redundant Representations: From Theory to Applications in Signal and Image Processing*, Springer, 2010.
- [10] G. Peyre, J. Fadili, and J. Starck, “Learning the morphological diversity,” *SIAM Journal of Imaging Science*, vol. 3, no. 3, pp. 646–669, 2010.
- [11] M. Elad and M. Aharon, “Image denoising via sparse and redundant representations over learned dictionaries,” *IEEE Trans. on Image Processing*, vol. 15, no. 12, pp. 3736–3745, December 2006.
- [12] M. Aharon, M. Elad, and A. Bruckstein, “K-SVD: An algorithm for designing overcomplete dictionaries for sparse representation,” *IEEE Trans. on Signal Processing*, vol. 54, no. 11, pp. 4311–4322, 2006.
- [13] J.-F. Cardoso, H. Snoussi, J. Delabrouille, and Patanchon G., “Blind separation of noisy Gaussian stationary sources. Application to cosmic microwave background imaging,” in *EUSIPCO Conference*, Toulouse, France, September 2002, pp. 561–564.
- [14] R. Rubinstein, M. Zibulevsky, and M. Elad, “Efficient implementation of the K-SVD algorithm using batch orthogonal matching pursuit,” Tech. Rep., Technion, 2008.