JOINT IMAGE SEPARATION AND DICTIONARY LEARNING

Xiaochen Zhao, Guangyu Zhou, Wei Dai
Department of Electrical and Electronic Engineering
Imperial College London
London, United Kingdom
{xiaochen.zhao10, g.zhou11, wei.dai1}@imperial.ac.uk

Tao Xu, Wenwu Wang
Department of Electronic Engineering
University of Surrey
Guildford, United Kingdom
{t.xu, w.wang}@surrey.ac.uk

***Abstract*—Blind source separation (BSS) aims to estimate unknown sources from their mixtures. Methods to address this include the benchmark ICA, SCA, MMCA, and more recently, a dictionary learning based algorithm BMMCA. In this paper, we solve the separation problem by using the recently proposed SimCO optimization framework. Our approach not only allows to unify the two sub-problems emerging in the separation problem, but also mitigates the singularity issue which was reported in the dictionary learning literature. Another unique feature is that only one dictionary is used to sparsely represent the source signals while in the literature typically multiple dictionaries are assumed (one dictionary per source). Numerical experiments are performed and the results show that our scheme significantly improves the performance, especially in terms of the accuracy of the mixing matrix estimation.**

***Index Terms*—Blind source separation, dictionary learning, image processing, optimization.**

I. INTRODUCTION

Blind source separation (BSS) has been investigated during the past two decades in a wide range of application fields such as speech and image separation. Early studies focus on the instantaneous and (over-)determined BSS problem, and address the problem under the framework of independent component analysis (ICA) [1], assuming that the sources are statistically independent. This has led to some well-known approaches, such as Infomax [2], maximum likelihood estimation [3], the maximum a posterior (MAP) [4], and FastICA [1]. Convolutive and/or underdetermined BSS problems have also been extensively studied especially in the speech processing applications, where the sensor measurements are usually modelled as convolutive (often underdetermined) mixtures of the original sources due to the presence of room reverberations (and often more sources than sensors). Effort in this direction has led to algorithms such as degenerate unmixed estimation technique (DUET) [5], non-negative matrix factorization (NMF) [6], and sparse representation technique [7].

In this paper, we focus on blind image separation application, in which the instantaneous model is usually adopted. To address this problem, several approaches have been proposed in the literature, including, for example, the Bayesian approaches based on Markov random field model (MRF) [8], sparse component analysis (SCA) [9] and morphological component analysis (MCA) [10] based on sparse representations. In MCA, source separation is addressed by decomposing the images into different morphological components in terms of sparsity of each component in a signal dictionary. The MCA has also been extended to multichannel case as multichannel MCA (MMCA) [11] and generalized MCA (GMCA) [12]. In MMCA, each source is assumed to be sparse in a specific transform domain. However, in GMCA, each source can be represented by the linear combination of morphological components and each component has a sparse representation by a specific dictionary. Recently, MMCA is further adapted to BMMCA [13] based on learned dictionary for separating mixed images. This method is motivated by the idea of image denoising using a learned dictionary from corrupted image in [14], which in principle extends the denoising problem to BSS. The BMMCA method is interesting in that the dictionary is directly trained from the mixtures, alleviating the issue of requiring training data, and as a result the algorithm can still perform in a blind manner. However, the BMMCA method trains multiple dictionaries for different sources, and in each iteration only updates one atom, rendering a potentially ineffective sparse representation of the image sources and a computationally inefficient procedure.

In this paper, we propose a new method, termed *SparseBSS*, which not only addresses the above limitations but also has some interesting new properties (discussed below). The implementation is based on simultaneous codeword optimization (SimCO) [15] framework on Grassmann manifolds, which ensures that the constraints on the column norms of the mixing matrix and dictionaries are satisfied. Numerical experiments for blind image separation show the advantages of SparseBSS over the ICA, GMCA, and BMMCA methods.

The major differences of our proposed algorithm from the existing methods include:

- Different from most dictionary based BSS algorithms where multiple dictionaries are used, we use only one dictionary to sparsely represent different sources. On one hand, this reduces the computational cost. On the other hand, there is no noticeable performance difference between the two approaches when the single dictionary used contains sufficient many codewords (the number of codewords is still less than that of multiple dictionaries combined).
- Formulating the overall separation problem into two sub-problems, we adapt the recently proposed SimCO optimization method [15] to solve both. The advantage of unifying the two stages is that, in practice, the same algorithm framework and codes can be used for both stages, thus significantly reducing the implementation effort.
- Another important reason to adapt the SimCO framework is to alleviate the possible ill-convergence problem ex-

isting in the traditional dictionary learning methods, e.g., K-SVD [16] and MOD [17]. In [15], it was observed that singular points, rather than the local minima, tend to be the major obstacle preventing algorithm from converging to a global minimizer. By adopting regularized SimCO, we are able to force the search path away from singular points and improve the performance.

The remainder of this paper is organized as follows. Section II is devoted to introduce our proposed framework. Then we briefly discussed some related methods in Section III. The algorithmic details of SparseBSS is presented in Section IV. The comparisons among the proposed algorithm, and benchmark methods ICA, GMCA, and BMMCA are analyzed and demonstrated in the last Section.

II. Problem Formulation

We first describe the mixing model. An image source $\mathcal{S}_i$ of size $\sqrt{N} \times \sqrt{N}$ is represented as a row vector $\boldsymbol{s}_i \in \mathbb{R}^{1\times N}$. Let the matrix $\boldsymbol{S} = \left[\boldsymbol{s}_1^T, \boldsymbol{s}_2^T, ..., \boldsymbol{s}_s^T\right]^T$ contain the s sources that will be separated. Let $\boldsymbol{Z} \in \mathbb{R}^{r\times N}$ denote the observed mixtures. Assume the linear mixing model

$$\boldsymbol{Z} = \boldsymbol{A}\boldsymbol{S} + \boldsymbol{V}, \tag{1}$$

where $\boldsymbol{A} \in \mathbb{R}^{r\times s}$ is referred to as the mixing matrix, and $\boldsymbol{V} \in \mathbb{R}^{r\times N}$ denote the i.i.d. additive Gaussian noise with mean zero and variance σ^2. In this work, we assume that the number of the sources s and the noise variance σ^2 are known.

The aim of BSS is to find both the mixing matrix $\boldsymbol{A}$ and the sources $\boldsymbol{S}$ from the observations $\boldsymbol{Z}$. The commonly used formulation is the following optimization problem:

$$\min_{\boldsymbol{A},\boldsymbol{S}} \left\|\boldsymbol{Z} - \boldsymbol{A}\boldsymbol{S}\right\|_F^2, \tag{2}$$

where $\left\|\cdot\right\|_F$ denotes the Frobenius norm. Note that this is an ill-posed problem: the separation result may not be meaningful unless extra constraints are introduced.

Our approach is based on the sparsity assumption. That is, the sources $\boldsymbol{S}$ can be sparsely represented under an over-complete dictionary $\boldsymbol{D}$, i.e., $\boldsymbol{S}^T \approx \boldsymbol{D}\boldsymbol{X}$ where most entries in $\boldsymbol{X}$ are zero. In the BSS problem, it is natural to assume that there is no prior information about both $\boldsymbol{D}$ and $\boldsymbol{X}$. The sparse representation problem, or the *dictionary learning problem*, can be written as

$$\min_{\boldsymbol{D},\boldsymbol{X}} \left\|\boldsymbol{X}\right\|_0 \; s.t. \, \boldsymbol{S}^T = \boldsymbol{D}\boldsymbol{X}, \tag{3}$$

where $\left\|\cdot\right\|_0$ is the ℓ_0 pseudo-norm which counts the number of non-zero components.

For the practice of image separation, multiple overlapped segments (patches) of the sources are used for dictionary learning in order to reduce the dimensionality. The linear mapping from a source image $\boldsymbol{s}_i$ to a patch can be described by a binary matrix $\boldsymbol{P} \in \mathbb{R}^{n\times N}$: the product $\boldsymbol{P}\cdot\boldsymbol{s}_i^T \in \mathbb{R}^{n\times 1}$ gives the vectorized version of a $\sqrt{n} \times \sqrt{n}$ patch from the image $\mathcal{S}_i$. Consider multiple patches (usually overlapped) and the corresponding patching operators $\boldsymbol{P}_1, \cdots, \boldsymbol{P}_K \in \mathbb{R}^{n\times N}$. Define $\boldsymbol{P} = [\boldsymbol{P}_1, ..., \boldsymbol{P}_K] \in \mathbb{R}^{n\times KN}$. Then the operation of extracting multiple patches from multiple sources $\boldsymbol{S}$ can be described by the linear operator $\mathcal{P}\boldsymbol{S} = ([\boldsymbol{P}_1, ..., \boldsymbol{P}_K]) \cdot (\left[\boldsymbol{s}_1^T, \boldsymbol{s}_2^T, ..., \boldsymbol{s}_s^T\right] \otimes \boldsymbol{I}_K) = \boldsymbol{P}\cdot(\boldsymbol{S}^T \otimes \boldsymbol{I}_K) \in \mathbb{R}^{n\times Ks}$, where the symbol $\otimes$ denotes the Kronecker product and $\boldsymbol{I}_K$ is the $K\times K$ identity matrix. Here, each column of $\mathcal{P}\boldsymbol{S}$ is one vectorized patch. The corresponding pseudo inverse $\mathcal{P}^\dagger$ is well defined. Due to the structure of $\mathcal{P}$, the computational costs of both $\mathcal{P}$ and $\mathcal{P}^\dagger$ are extremely low. With the above definitions, the dictionary learning problem is typically formulated as

$$\min_{\boldsymbol{D},\boldsymbol{X}} \left\|\boldsymbol{D}\boldsymbol{X} - \mathcal{P}\boldsymbol{S}\right\|_F^2 + \mu\left\|\boldsymbol{X}\right\|_0. \tag{4}$$

With the sparsity constraint involved, the BSS problem can be then written as a joint optimization

$$\min_{\boldsymbol{A},\boldsymbol{S},\boldsymbol{D},\boldsymbol{X}} \lambda\left\|\boldsymbol{Z} - \boldsymbol{A}\boldsymbol{S}\right\|_F^2 + \left\|\boldsymbol{D}\boldsymbol{X} - \mathcal{P}\boldsymbol{S}\right\|_F^2 + \mu\left\|\boldsymbol{X}\right\|_0, \tag{5}$$

where the parameter λ is introduced to balance the measurement error and the sparse approximation error. It is clearly difficult to solve (5). To simplify the problem, we divide the overall optimization problem into two sub-problems (two stages):

- Dictionary learning stage (with $\boldsymbol{A}$ and $\boldsymbol{S}$ fixed)

$$\min_{\boldsymbol{D},\boldsymbol{X}} \left\|\boldsymbol{D}\boldsymbol{X} - \mathcal{P}\boldsymbol{S}\right\|_F^2 + \mu\left\|\boldsymbol{X}\right\|_0, \tag{6}$$

- Mixture learning stage (with $\boldsymbol{D}$ and $\boldsymbol{X}$ fixed)

$$\min_{\boldsymbol{A},\boldsymbol{S}} \lambda\left\|\boldsymbol{Z} - \boldsymbol{A}\boldsymbol{S}\right\|_F^2 + \left\|\boldsymbol{D}\boldsymbol{X} - \mathcal{P}\boldsymbol{S}\right\|_F^2. \tag{7}$$

The overall separation process involves iteratively solving these two sub-problems until convergence.

In our approach, further constraints are posed on the feasibility of $\boldsymbol{A}$ and $\boldsymbol{D}$. In particular, we require each column of $\boldsymbol{A}$ and $\boldsymbol{D}$ of unit ℓ_2-norm. The feasible sets can be then written as

$$\mathcal{A} = \left\{\boldsymbol{A} \in \mathbb{R}^{r\times s} : \left\|\boldsymbol{A}_{:,i}\right\|_2 = 1, \; 1 \le i \le s\right\} \tag{8}$$

and

$$\mathcal{D} = \left\{\boldsymbol{D} \in \mathbb{R}^{n\times d} : \left\|\boldsymbol{D}_{:,i}\right\|_2 = 1, \; 1 \le i \le d\right\}, \tag{9}$$

respectively. Such constraints were used in blind source separation and dictionary learning literature before as the corresponding optimization problems (2) and (4) are invariant to column scaling. Here, however, the constraints make a difference when the joint optimization (5) is considered. For example, let us fix $\boldsymbol{A}$, $\boldsymbol{S}$, $\boldsymbol{D}$, $\boldsymbol{X}$ and consider the scaling version $c\boldsymbol{A}$, $c^{-1}\boldsymbol{S}$, $\boldsymbol{D}$, $c^{-1}\boldsymbol{X}$ for some $c > 0$. The objective function in (5) becomes $\lambda\left\|\boldsymbol{Z} - \boldsymbol{A}\boldsymbol{S}\right\|_F^2 + c^{-2}\left\|\boldsymbol{D}\boldsymbol{X} - \mathcal{P}\boldsymbol{S}\right\|_F^2 + \mu\left\|\boldsymbol{X}\right\|_0$. If $c > 1$ keeps increasing, then the objective function keeps decreasing accordingly. At the same time, the solutions are essentially the same. Adding constraints (8) and (9) helps in avoiding the above problem. Additional advantages of the constraints in the dictionary learning stage are detailed in [15] and omitted here. The algorithmic details on how to address these constraints in the optimization process are presented in Section IV.

It is also worth to note that only one dictionary is used in the proposed SparseBSS approach. This is different from the MMCA and BMMCA methods, described in details in Sections III-A and III-B respectively, where one source corresponds one dictionary. The advantage of single dictionary is that the complexity will typically increase sub-linearly as the number of sources increases. In the numerical test (Section V), the single dictionary used contains codewords less than the codewords (combined from multiple dictionaries) used in MMCA and BMMCA. Besides the reduced computational complexity, performance improvement can be observed.

III. Related Work

To better differentiate the proposed SparseBSS method, in this section we briefly discuss the MMCA and BMMCA algorithms, both of which rely on the sparsity assumption as well. The popular ICA approach is not detailed here as it stands for a totally different approach. Interested readers may refer to [1] for more details.

A. Multichannel MCA for Blind Source Separation

In MMCA [11], each source $\boldsymbol{s}_i$ is assumed to be sparsely represented by a different dictionary $\boldsymbol{D}_i$, which is known a priori or trained from the source, i.e., $\boldsymbol{s}_i = \boldsymbol{D}_i \boldsymbol{x}_i$ and $\boldsymbol{x}_i$ is sparse. Hence, the optimization formulation becomes

$$\min_{\boldsymbol{A},\boldsymbol{S}} \|\boldsymbol{Z} - \boldsymbol{A}\boldsymbol{S}\|_F^2 + \sum_{i=1}^{n} \lambda_i \left\|\boldsymbol{s}_i \boldsymbol{D}_i^{\dagger}\right\|_1 , \tag{10}$$

where $\|\cdot\|_1$ denotes the ℓ_1-norm and is used here to promote sparsity, and the weighting parameter $\lambda > 0$ is used to balance the sparsity and estimation error. The main drawback of MMCA is the assumption that the dictionaries for the sources have to be known a priori, which is often not satisfied in practical scenarios.

B. BMMCA

To avoid the major limitation of MMCA, the BMMCA approach in [13] proposes to learn the dictionaries during the separation process. To reduce the dimensionality, BMMCA also divide the image sources into patches. Let $\mathcal{R}$ be the linear operator to extract patches from an image source and stack the patches into a matrix. (We use a different notation than the previous patching operator $\mathcal{P}$ because $\mathcal{R}$ handles a single source and $\mathcal{P}$ deals with multiple sources.) Let $\boldsymbol{D}_i$ be the dictionary corresponding to the i^{th} source and $\boldsymbol{X}_i$ be the corresponding sparse coefficients. The optimization formulation in BMMCA is given by

$$\min_{\boldsymbol{A}_{:,i},\boldsymbol{s}_i,\boldsymbol{D}_i,\boldsymbol{X}_i} \lambda \|\boldsymbol{E}_i - \boldsymbol{A}_{:,i}\boldsymbol{s}_i\|_F^2 + \|\boldsymbol{D}_i\boldsymbol{X}_i - \mathcal{R}\boldsymbol{s}_i\|_2^2 + \mu \|\boldsymbol{X}_i\|_0 , \tag{11}$$

where the matrix $\boldsymbol{E}_i$ is defined as

$$\boldsymbol{E}_i = \boldsymbol{Z} - \sum_{j \neq i} \boldsymbol{A}_{:,j}\boldsymbol{s}_j ,$$

and i varies from 1 to s. To learn the dictionary $\boldsymbol{D}_i$ from the source $\boldsymbol{s}_i$, BMMCA adopts the K-SVD method [16].

Though the optimization formulation (11) is quite similar to our SparseBSS formulation in (5), these two methods differ in the following aspects. First, BMMCA assumes different dictionaries for different sources while SparseBSS uses a single dictionary for all the sources. Second, BMMCA employs the K-SVD mechanism for both dictionary learning and mixing learning stages while SparseBSS relies on the SimCO framework for both. It is worth to note that K-SVD was designed to solve the problem in the form of (2). Applying it to the optimization (7) is troublesome: the claimed optimality in each individual step will not hold any more due to the second term in (7). By contrast, the SimCO framework can be readily applied to both (2) and (7).

IV. Algorithmic Details

Our model provides an alternating joint update between $\{\boldsymbol{D}, \boldsymbol{X}\}$ and $\{\boldsymbol{A}, \boldsymbol{S}\}$. Different from the current benchmark image separation algorithms, we train only one dictionary instead of multiple dictionaries in our separation model. Another difference is that we adapt regularized SimCO framework [15] to unify the two stages of the algorithm, i.e., both mixture learning and dictionary learning stage can be updated by using the same optimization framework. Such algorithm design significantly reduces the implementation efforts. More importantly, it provides a way to better alleviate the singularity problem [15] and hence result in better separation results.

We discuss the two optimization stages in the following two subsections, respectively, then present the proposed algorithm at the end of this section.

A. Dictionary learning stage

Typically, most dictionary learning algorithm contains two steps: sparse coding and dictionary update. In sparse coding step, we solve the following least squares problem,

$$\min_{\boldsymbol{X}} \|\boldsymbol{X}\|_0 \text{ s.t. } \|\boldsymbol{D}\boldsymbol{X} - \mathcal{P}\boldsymbol{S}\|_F \leq \epsilon, \tag{12}$$

by keeping the dictionary fixed and using some sparse coding techniques, e.g., orthogonal matching pursuit (OMP) [18], subspace pursuit (SP) [19] to update $\boldsymbol{X}$. The constant ϵ is an error bound being proportional to the noise standard deviation. Sparse coding algorithm outputs the sparse coefficients $\boldsymbol{X}$ and its sparse pattern $\Omega = \{(i,j) : \boldsymbol{X}(i,j) \neq 0\}$, which contains the positions of non-zero elements in $\boldsymbol{X}$. For the dictionary update step, there are several methods, e.g., MOD, K-SVD and SimCO. In MOD, one keeps the updated sparse coefficient $\boldsymbol{X}$ fixed and search for the optimal dictionary $\boldsymbol{D}$. For the K-SVD algorithm, which is used in BMMCA, one updates one codeword of $\boldsymbol{D}$ and the the corresponding row of the sparse coefficient matrix $\boldsymbol{X}$ until all the codewords and the corresponding sparse coefficients been updated eventually. For the proposed algorithm, we consider the SimCO approach, which is designed to simultaneously train all the codewords of the dictionary and the sparse coefficients from the given data, which can be sparsely represented. Note that we keep the sparse pattern Ω fixed at dictionary update step.

In all above three dictionary update methods, an ill-conditioned dictionary may prevent the optimization process approaching to the global minimizer. The convergence analysis in [15] shows that the failure of finding a global minimizer is mainly due to the singularity of the updated dictionary. For this reason, a regularized term $\|\boldsymbol{X}\|_F^2$ is added in the

SimCO formulation to improve its learning performance. The regularized SimCO, in the dictionary update stage is therefore formulated as

$$\min_{\boldsymbol{D}\in\mathcal{D}} \quad \min_{\boldsymbol{X}\in\Omega} \left\|\boldsymbol{DX}-\mathcal{P}\boldsymbol{S}\right\|_F^2+\mu\left\|\boldsymbol{X}\right\|_F^2, \tag{13}$$

$$= \min_{\boldsymbol{D}\in\mathcal{D}} \underbrace{\min_{\boldsymbol{X}\in\Omega} \left\|\begin{bmatrix}\boldsymbol{D}\\ \sqrt{\mu}\boldsymbol{I}\end{bmatrix}\boldsymbol{X}-\begin{bmatrix}\mathcal{P}\boldsymbol{S}\\ \boldsymbol{0}\end{bmatrix}\right\|_F^2}_{f(\boldsymbol{D})}, \tag{14}$$

$$= \min_{\boldsymbol{D}\in\mathcal{D}} \underbrace{\min_{\boldsymbol{X}\in\Omega} \left\|\tilde{\boldsymbol{D}}\boldsymbol{X}-\tilde{\boldsymbol{S}}\right\|_F^2}_{f(\boldsymbol{D})} \tag{15}$$

where $\mu > 0$ is a regularization parameter. In real data testing, we found that the regularized SimCO gives better performance than the other dictionary learning algorithms (such as K-SVD and MOD) when applied to the image denoising problem [15]. For this reason, regularized SimCO is applied in our dictionary learning stage.

B. Mixture learning stage

In this stage, we simultaneously update the mixing matrix $\boldsymbol{A}$ and the sources $\boldsymbol{S}$. Consider formulation (7). It contains a summation of two terms both of the form $\left\|\boldsymbol{C}_1-\boldsymbol{C}_2\boldsymbol{S}\right\|_F^2$ as Equation (14). Therefore similar method in SimCO can also be used in solving this problem. Referring to (14), denote

$$\tilde{\boldsymbol{Z}}=\begin{bmatrix}\sqrt{\lambda}\boldsymbol{Z}\\ \mathcal{P}^{\dagger}\left(\boldsymbol{DX}\right)\end{bmatrix}, \tilde{\boldsymbol{A}}=\begin{bmatrix}\sqrt{\lambda}\boldsymbol{A}\\ \boldsymbol{I}\end{bmatrix}, \tag{16}$$

where $\mathcal{P}^{\dagger}$ denotes a Moore-Penrose pseudo-inverse operator of $\mathcal{P}$ and $\boldsymbol{DX}$ is obtained in the dictionary learning stage and fixed in this stage. $\mathcal{P}^{\dagger}\left(\boldsymbol{DX}\right)$ recovers the estimated patches $\boldsymbol{DX}$ to the estimated vectorized sources. Note that $\mathcal{P}^{\dagger}$ is easy to compute due to the structure of $\mathcal{P}$. The problem formulation (7) of the mixture learning stage can therefore be written as

$$\min_{\boldsymbol{A},\boldsymbol{S}} \lambda\left\|\boldsymbol{Z}-\boldsymbol{AS}\right\|_F^2+\left\|\boldsymbol{DX}-\mathcal{P}\boldsymbol{S}\right\|_F^2$$

$$= \min_{\boldsymbol{A}\in\mathcal{A}} \underbrace{\min_{\boldsymbol{S}} \left\|\begin{bmatrix}\sqrt{\lambda}\boldsymbol{Z}\\ \mathcal{P}^{\dagger}\left(\boldsymbol{DX}\right)\end{bmatrix}-\begin{bmatrix}\sqrt{\lambda}\boldsymbol{A}\\ \boldsymbol{I}\end{bmatrix}\boldsymbol{S}\right\|_F^2}_{f(\boldsymbol{A})}$$

$$= \min_{\boldsymbol{A}\in\mathcal{A}} \underbrace{\min_{\boldsymbol{S}} \left\|\tilde{\boldsymbol{Z}}-\tilde{\boldsymbol{A}}\boldsymbol{S}\right\|_F^2}_{f(\boldsymbol{A})} \tag{17}$$

Hence, one can directly apply the SimCO mechanism to the mixture learning stage. We will discuss how to simultaneously update the mixing matrix $\boldsymbol{A}$ and the sources $\boldsymbol{S}$ in the next subsection.

C. Line Search Method

During the update of dictionary update stage and mixture learning stage, we both use gradient descent method. For the sake of space, we only discuss how to simultaneously update $\{\boldsymbol{A}, \boldsymbol{S}\}$ as it is the same to update $\{\boldsymbol{D}, \boldsymbol{X}\}$ in dictionary learning stage. With the notation $f(\boldsymbol{A})$ in (17), we treat $\boldsymbol{S}$ as a function of $\boldsymbol{A}$. For a given $\boldsymbol{A}$, the corresponding optimal $\boldsymbol{S}^*$ can be computed as

$$\boldsymbol{S}^*=\tilde{\boldsymbol{A}}^{\dagger}\tilde{\boldsymbol{Z}}, \tag{18}$$

where $\tilde{\boldsymbol{A}}^{\dagger}$ is the pseudo-inverse of $\tilde{\boldsymbol{A}}$. Then,

$$\begin{aligned}\nabla_{\boldsymbol{A}}f &= \frac{\partial f}{\partial \boldsymbol{A}}\mid_{\boldsymbol{S}^*}+\frac{\partial f}{\partial \boldsymbol{S}}\mid_{\boldsymbol{S}^*}\cdot\frac{\partial \boldsymbol{S}}{\partial \boldsymbol{A}}\\ &= \frac{\partial f}{\partial \boldsymbol{A}}\mid_{\boldsymbol{S}^*}+\boldsymbol{0}\cdot\frac{\partial \boldsymbol{S}}{\partial \boldsymbol{A}}\\ &= -2\left(\tilde{\boldsymbol{Z}}-\tilde{\boldsymbol{A}}\boldsymbol{S}^*\right)\boldsymbol{S}^{*T}.\end{aligned} \tag{19}$$

Notice that both the dictionary $\boldsymbol{D}$ and the mixing matrix $\boldsymbol{A}$ are constrained to have unit column norms. Common selection of an updated direction probably result in an updated $\boldsymbol{D}\notin\mathcal{D}$ or $\boldsymbol{A}\notin\mathcal{A}$. As discussed in last subsection, such results tremendously increase the solution set of the optimal dictionaries and the mixing matrix which probably cause the failure of the algorithm. Thereafter we refer to [20][21] and restrict the line search path to the product of Grassmann manifolds. To reach this, we define a projection operator $\bar{(\cdot)}$. Let $\boldsymbol{u}\in\mathcal{G}$, where the Grassmann manifold $\mathcal{G}=\{span(\boldsymbol{u}):\ \boldsymbol{u}\in\mathcal{U}\}$ and Stiefel manifold $\mathcal{U}=\left\{\boldsymbol{u}\in\mathbb{R}^m:\ \boldsymbol{u}^T\boldsymbol{u}=1\right\}$.

$$\bar{\boldsymbol{h}}=\left(\boldsymbol{I}-\boldsymbol{u}\boldsymbol{u}^T\right)\boldsymbol{h}, \tag{20}$$

so that $\bar{\boldsymbol{h}}$ and $\boldsymbol{u}$ are orthogonal. Here $\boldsymbol{u}$ can be a codeword in dictionary $\boldsymbol{D}$ or a column of mixing matrix $\boldsymbol{A}$. Vector $\boldsymbol{h}$ represents $\nabla_{\boldsymbol{A}_{:,i}}f$, $i\in[s]$ in the mixture learning stage and $\nabla_{\boldsymbol{D}_{:,j}}f$, $j\in[d]$ in the dictionary learning stage. For a given non-zero direction $\bar{\boldsymbol{h}}$ and a step size $t\in\mathbb{R}$, $\boldsymbol{u}$ is updated as

$$\boldsymbol{u}\left(t\right)=\boldsymbol{u}\cdot\cos\left(\left\|\bar{\boldsymbol{h}}\right\|_2 t\right)+\frac{\bar{\boldsymbol{h}}}{\left\|\bar{\boldsymbol{h}}\right\|_2}\cdot\sin\left(\left\|\bar{\boldsymbol{h}}\right\|_2 t\right). \tag{21}$$

A pseudcode of the proposed algorithm is given in Algorithm 1. Step 1 is the dictionary learning stage and Step 2 is the mixture learning stage. In the dictionary learning stage, the estimated sources are sparsely represented by the dictionary and the sparse coefficients. In the mixture learning stage, the dictionary and sparse coefficients are used to update the estimated sources and then find a good mixing matrix using gradient descent method. We iteratively repeat these two steps until a convergence.

V. SIMULATIONS

In the simulations, two source images were mixed together using a 4 × 2 full rank random column normalized mixing matrix $\boldsymbol{A}$. Normally, the patch size depends on the size of the sources. We chose 8 × 8 patches from the source images with size $N = 128\times 128$. The overlap percentage of the patches was fixed to 50% for our proposed algorithm in both noise and noiseless cases. In order to pursuit a good recovery result, it is better to keep the overlap percentage at a high level. For the dictionary learning stage, we would like to emphasize again that only one dictionary was generated to sparsely represent all source images. The number of atoms of the dictionary was $d = 256$. We will not discuss the optimum dictionary redundancy factor d/n as it is beyond the scope of this paper. Based on the experiments in [15],

Algorithm 1 SparseBSS Algorithm.

```
Input: Observations $\boldsymbol{Z}$, patch size $n$, number of dictionary codewords $d$, regularization parameters $\lambda$ and $\mu$, and total number of iterations $l_{max}$.
Output: Dictionary $\boldsymbol{D}$, sparse coefficients $\boldsymbol{X}$, separated images $\boldsymbol{S}$, and estimated mixing matrix $\boldsymbol{A}$.
Initialization: Set $\boldsymbol{D}$ to over-complete DCT. Set a random column-normalized matrix $\boldsymbol{A}$. Compute $\boldsymbol{S} = \boldsymbol{A}^{\dagger}\boldsymbol{Z}$.
For $k = 1, 2, \ldots, l_{max}$ repeat step $(1) - (2)$.
    1) Dictionary learning stage
        Sparse coding $\boldsymbol{X} \leftarrow \arg\min_{\boldsymbol{X}} \|\boldsymbol{DX} - \mathcal{P}\boldsymbol{S}\|_F^2$.
        Update $\boldsymbol{D}, \boldsymbol{X} \leftarrow \arg \min_{\boldsymbol{D} \in \mathcal{D}, \boldsymbol{X} \in \Omega} \|\boldsymbol{DX} - \mathcal{P}\boldsymbol{S}\|_F^2 + \mu \|\boldsymbol{X}\|_F^2$.
    2) Mixture learning stage
        Compute $\boldsymbol{S} = \tilde{\boldsymbol{A}}^{\dagger}\tilde{\boldsymbol{Z}}$.
        Update $\boldsymbol{A} \leftarrow \arg\min_{\boldsymbol{A} \in \mathcal{A}} \left\|\tilde{\boldsymbol{Z}} - \tilde{\boldsymbol{A}}\boldsymbol{S}\right\|_F^2$.
end
```

Table I
ACHIEVED MSES OF THE ALGORITHMS IN NOISELESS CASE.

	FastICA	GMCA	BMMCA	SparseBSS
Lena	8.7489	4.3780	3.2631	**3.1346**
Boat	18.9269	**6.3662**	12.5973	6.6555

the parameter μ of the penalty term was fixed to 0.05. For the mixture learning stage, the constant parameter λ depends on specific noise level of the observations. For SparseBSS, the total number of iterations l_{max} was fixed to 50. Each iteration consists of one implementation of dictionary learning and five implementations of mixture learning. Consequently, all above parameters were fixed in the experiments, except for the parameter λ.

For the first experiment, we selected two classic images, *Lena* and *Boat* as the source images, which are shown in Fig. 2 (a). We compared SparseBSS with other benchmark algorithms: FastICA[1] [22], GMCA[2] [12] and BMMCA [13]. For the noiseless case, we calculate the Mean Square Errors (MSEs) to compare the reconstruction performance of the candidate algorithms. The lower the MSE, the better the reconstruction performance. MSE is given in $MSE = (1/N)\|\chi - \tilde{\chi}\|_F^2$, where χ is the source image and $\tilde{\chi}$ is the reconstructed image. For the BMMCA algorithm, we set the total number of iterations to be 500 and the overlap percentage was 50%. Table 1 illustrates the results of four tested algorithms. GMCA and Sparse had similar results for boat. However, for Lena, ours is better than GMCA and BMMCA. The results of FastICA is not as good as those three algorithms. For the noise case, we also tested those four algorithms. In this case, we added Gaussian white noise with σ equaling to 10 to the four mixtures, which is shown in Fig. 1. The Peak Signal-to-Noise Ratio (PSNR) is used as a measurement of the reconstruction quality. Better quality leads to higher PSNR. It is defined as, $PSNR = 20\log_{10}(\frac{MAX}{\sqrt{MSE}})$, where MAX indicates the maximum possible pixel value of the image. For a uint-8 image, the MAX equals to 255. The separation results are shown in Fig. 2 (b)-(e). For the BMMCA algorithm, 200 iterations were set as the stopping criterion and full overlapped patches were selected, which increased the computational complexity. All algorithms successfully separated the noise mixtures. However, FastICA algorithm fails to denoise and GMCA blurred the images. The results of BMMCA are smooth but lost lots of image details. SparseBSS offer significant performance improvement in both separation and denoising, *e.g. Lena*'s facial details are the most legible among the four. Moreover, it is mentioned that the overlap percentage of the patches of SparseBSS was fixed to 50%. Lower overlap percentage will speed up the separation process, while a higher one will bring better separation performance.

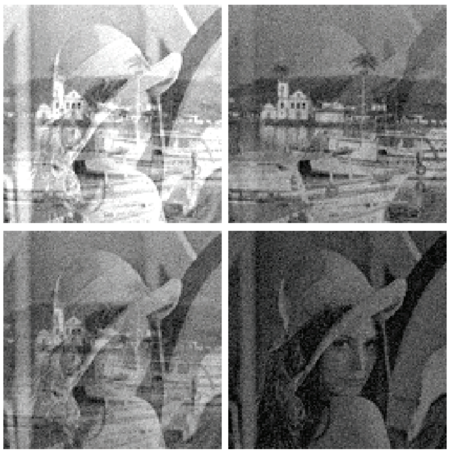

Figure 1. Four noisy mixtures with Gaussian noise ($\sigma = 10$).

It is also worth mentioning about the learned dictionary from the mixtures. After applying the SimCO algorithm, the trained dictionary surprisingly looks like the initialization in which an over-complete DCT dictionary was used. Similar dictionaries are also trained for image denoising solved by SimCO [15]. This is quite different from the ones trained via K-SVD algorithm [14]. However dictionaries trained via SimCO can represent images with the same sparsity level as the ones trained via K-SVD and also reach very similar performance. We are doing more detailed analysis on this problem. Note that an over-complete DCT dictionary can already sparsely represent images, therefore the trained dictionary from SparseBSS is a reasonable solution.

At last, we compared the performance of all the methods in different noise levels. We use the mixing matrix error as the measurement of the performance. The *mixing matrix error* is defined as $E_{\boldsymbol{A}} = \left\|\boldsymbol{A} - \hat{\boldsymbol{A}}\right\|_F^2$, where $\hat{\boldsymbol{A}}$ is the approximated column normalized and reformulated mixing matrix. In this experiment, the noise level, which is also the noise standard deviation, varies from 2 to 20. The resulted curves are shown in Fig. 3. The performance of GMCA is better than that of FastICA. The curve for BMMCA is not available as the setting for the parameters are sophisticated and varies in different noise levels. It is hard to obtain a good result for BMMCA. SparseBSS outperforms the compared algorithms at all the

[1] Available at: http://research.ics.aalto.fi/ica/fastica/index.shtml

[2] Available at: http://md.cosmostat.org/Generalized_MCA.html

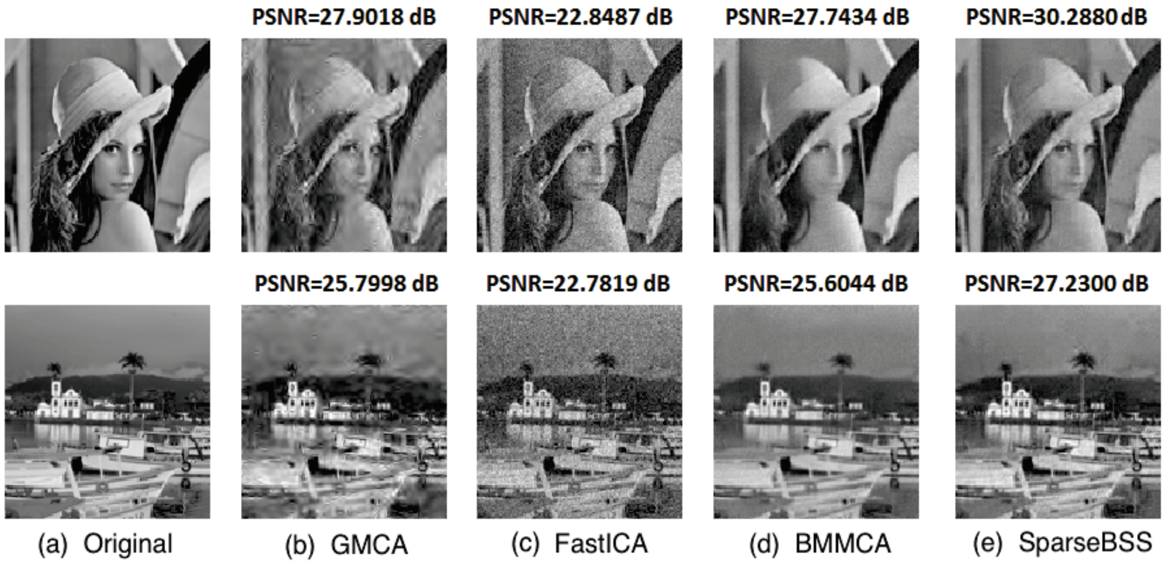

Figure 2. (a) Original Images. (b)∼(e) are separation results by GMCA, FastICA, BMMCA, and SparseBSS, respectively.

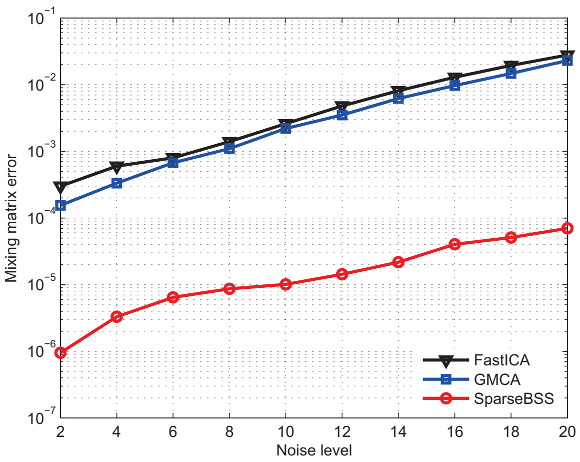

Figure 3. The performance of the tested algorithms at different noise levels.

tested noise levels.

References

[1] A. Hyvärinen, J. Karhunen, and E. Oja, *Independent Component Analysis*. New York: Wiley-Interscience, May 2001.

[2] J. Bell and T. J. Sejnowski, "An information-maximization approach to blind separation and blind deconvolution," *Neural Computation*, vol. 7, pp. 1129–1159, 1995.

[3] M. Gaeta and J. L. Lacoume, "Source separaion without prior knowledge: the maximum likelihood solution," in *Proceedings of EUSIPCO'90*, pp. 621–624, 1990.

[4] A. Belouchrani and J. F. Cardoso, "Maximum likelihood source separation for discrete sources," in *Proceedings of EUSIPCO*, pp. 768–771, 1994.

[5] A. Jourjine, S. Rickard, and O. Yilmaz, "Blind separation of disjoint orthogonal signals: Demixing N sources from 2 mixtures," in *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing*, pp. 2985–2988, 2000.

[6] D. D. Lee and H. S. Seung, "Learning the parts of objects by nonnegative matrix factorization," *Nature*, vol. 401, pp. 788–791, 1999.

[7] M. Zibulevsky and B. A. Pearlmutter, "Blind source separation by sparse decomposition," *Neural Computation*, vol. 13, pp. 863–882, 2001.

[8] K. Kayabol, E. E. Kuruoglu, and B. Sankur, "Bayesian separation of images modeled with MRFs using MCMC," *IEEE Transactions on Image Processing*, vol. 18, no. 5, pp. 982–994, 2009.

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

[10] J. Starck, M. Elad, and D. Donoho, "Redundant multiscale transforms and their application for morphological component analysis," *Advances in Imaging and Electron Physics*, vol. 132, pp. 287–348, 2004.

[11] 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, 2006.

[12] J. Bobin, J. Starck, J. Fadili, and Y. Moudden, "Sparsity and morphological diversity in blind source separation," *IEEE transactions on Image Processing*, vol. 16, no. 11, pp. 2662–2674, 2007.

[13] V. Abolghasemi, S. Ferdowsi, and S. Sanei, "Blind separation of image sources via adaptive dictionary learning," *IEEE Transactions on Image Processing*, vol. 21, no. 6, pp. 2921–2930, 2012.

[14] M. Elad and M. Aharon, "Image denoising via sparse and redundant representations over learned dictionaries," *IEEE Transactions on Image Processing*, vol. 15, pp. 3736–3745, Dec. 2006.

[15] W. Dai, T. Xu, and W. Wang, "Simultaneous codeword optimization (SimCO) for dictionary update and learning," *IEEE Transactions on Signal Processing*, vol. 60, pp. 6340–6353, Dec. 2012.

[16] M. Aharon, M. Elad, and A. Bruckstein, "K-SVD: An algorithm for designing overcomplete dictionaries for sparse representation," *IEEE Transactions on Signal Processing*, vol. 54, no. 11, pp. 4311–4322, 2006.

[17] K. Engan, S. O. Aase, and J. H. Husoy, "Method of optimal directions for frame design," in *IEEE International Conference on Acoustics, Speech, and Signal Processing*, vol. 5, pp. 2443–2446, 1999.

[18] Y. C. Pati, R. Rezaiifar, and P. S. Krishnaprasad, "Orthogonal matching pursuit: recursive function approximation with applications to wavelet decomposition," *IEEE Asilomar Conference on Signals, Systems and Computers*, pp. 40–44, 1993.

[19] W. Dai and O. Milenkovic, "Subspace pursuit for compressive sensing signal reconstruction," *IEEE Transactions on Information Theory*, vol. 55, pp. 2230–2249, May 2009.

[20] A. Edelman, T. A. Arias, and S. T. Smith, "The geometry of algorithms with orthogonality constraints," *SIAM J. Matrix Anal. Appl.*, vol. 20, pp. 303–353, 1999.

[21] W. Dai, E. Kerman, and O. Milenkovic, "A geometric approach to low-rank matrix completion," *IEEE Transactions on Information Theory*, vol. 58, pp. 237–247, 2011.

[22] A. Hyvarinen, "Fast and robust fixed-point algorithms for independent component analysis," *IEEE Transactions on Neural Networks*, vol. 10, pp. 626–634, May 1999.