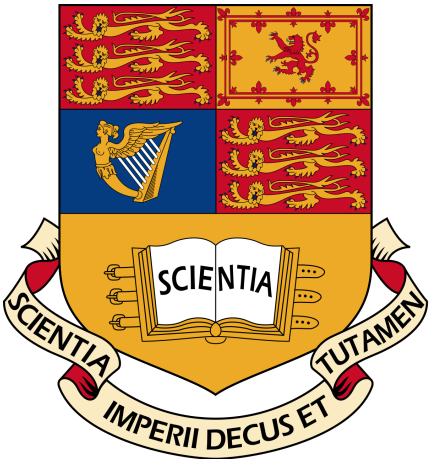
Imperial College London

Department of Electrical and Electronic Engineering

Final Year Project Report 2019



Project Title:	**Blind Source Separation Based on Sparsity**
Student:	**Zhongxuan Li**
CID:	**01050018**
Course:	**EE4**
Project Supervisor:	**Dr Wei Dai**
Second Marker:	**Dr Deniz Gunduz**

Abstract

Blind source separation (BSS) is a prevalent technique in array processing and data analysis that aims to recover unknown sources from observed mixtures, where the mixing matrix is unknown. The classical ICA methods require prior statistical knowledge that the sources have to be mutually independent. In order to overcome the limitations, sparsity based methods are introduced, which decompose the source signal sparsely in a prescribed dictionary. Morphological component analysis (MCA) theory is proposed based on theory of sparse representation. It assumes that the signal is a linear combination of several components having different geometry, and each embodiment of the component can be sparsely represented in a dictionary, and not sparsely represented in others. In recent years, this theory has been applied to solve the blind source separation problem and obtained good results.

The objectives of this report are to review some of the key approaches derived from the classical ICA methods that have been developed to address the BSS problem, and to further discuss sparsity based methods in blind source separation. It first describes the theory behind sparse representation and sparse decomposition algorithms, after which this report gives a decomposition algorithm based on block coordinate relaxation morphological component analysis whose variants have been applied to the multichannel morphological component analysis (MMCA) and generalised morphological component analysis (GMCA). A local dictionary learning (K-SVD) BSS algorithm is followed. Finally we improve the K-SVD BSS algorithm by further learning a block sparsifying dictionary (SAC+BK-SVD), which clusters the dictionary atoms according to their similarity and those atoms are updated by blocks.

In the implementation part, we are expected to perform image segmentation experiment and blind image source separation experiment using the techniques we have introduced. Another experiment involves comparing the proposed block-sparse dictionary learning algorithm with the K-SVD algorithm. Simulation results show the proposed methods yields better blind image separation quality.

Contents

1 Introduction

1.1 BSS preview

Imagine that two people are speaking simultaneously in a room. There are two microphones which are in different locations and record the stereo signals generated by two people. Assuming that the each of the recorded time signals $x_1(t)$ and $x_2(t)$ is a linear combination of the speeches $s_1(t)$ and $s_2(t)$. We could express this as a set of linear equations:

$$x_1(t) = a_{11}s_1(t) + a_{12}s_2(t) \tag{1}$$

$$x_2(t) = a_{21}s_1(t) + a_{22}s_2(t) \tag{2}$$

where parameter a_{ij} depends on the distance of the microphones from the speakers. Time delays or other extra factors are ignored temporarily from our simplified mixing model. It would be useful in real-world applications if we can estimate the original speech signals $s_1(t)$ and $s_2(t)$ using only the recorded signals $x_1(t)$ and $x_2(t)$. This is addressed as the well-known cocktail party problem and also the chief motivation behind blind source separation. The similar situation is also common in telecomms, medical signal and image processing. If we knew the parameters a_{ij} , we find $s_1(t)$ and $s_2(t)$ by solving linear equations or matrix inversion. The problem is, however we have no information about a_{ij}. the problem is considerable difficult.

The lack of prior knowledge of the mixing process can be compensated by a statistically strong but often physical plausible assumption of independence between the source signals. Independent Component Analysis (ICA) was first proposed to solve the cocktail party problem. The goal of ICA is to determine the original sources given mixtures of those sources, assuming that the sources are statistically independent and non Gaussian. Derivatives of the classical ICA methods includes JADE [7], FastICA [14]. Generally speaking, ICA algorithms are about devising adequate contrast functions which are related to approximation of independence[15]. However ICA is limited to the determined BSS problem when we have equal number of mixtures and the number of sources. This is because we need to find the inverse of the mixing matrix while optimising the contrast function in ICA. But only square matrix has such an inverse.

1.2 BSS by sparsity

Although ICA is proved to be effective in many BSS applications. The statistical independence assumption in the time domain cannot be applied to all scenarios. Sparsity-based approaches have drawn much attention in recent years. The term sparse refers to signals with small number of nonzeros with respect to some representation bases [26]. More specifically, sources have mutually disjoint support sets in a dictionary. This is exploited for instance in sparse component analysis (SCA) [13]. In SCA we make assumption that the sources to be unmixed can be sparsely represented in a predefined common basis or dictionary (for instance, a wavelet frame). A two-step approach [6] was proposed to solve the BSS problem using sparsity, in which the mixing system is first estimated using clustering methods, then the sources are estimated thanks to pursuit methods (e.g. basis pursuit, matching pursuit).

In many cases, basis pursuit or matching pursuit synthesis algorithms are computationally quite expensive. Furthermore, the traditional SCA requires highly sparse signals. Unfortunately, this is not the case for high dimensional signals and especially in image processing. We present in this report an alternative to these approaches, the morphological component analysis (MCA) [5, 4] is a method which sources can be sparsely represented using several different dictionaries. For example, images normally contains contour and texture, the former is well sparsified using curvelets transform whereas the latter may be well represented using local cosine transform (DCT). Multichannel morphological component analysis (MMCA) [22] and generalised morphological component analysis (GMCA) are extentions of MCA to the multichannel case. In MMCA setting, we assume that the sources have strictly different morphologies (i.e., each source is assumed to be sparsely representated in one particular orthonormal basis). In GMCA, each source is modeled as the linear combination of a number of morphologial components where each component is sparse in a specific orthonormal basis.

1.3 Structure of report

In next chapter, we first provide sufficient background knowledge for the blind source separation problem setting. Real world applications of BSS will be introduced before the performance measurements to evaluate different BSS algorithms are defined. Moreover, we will discuss the the well-established ICA algorithm as it is selected as our baseline method. We then turn our discussion to sparsity and morphological diversity. The idea of overcomplete dictionaries is followed. We will look in to multichannel morphological component Analysis and generalised morphological component analysis. In addition blind source separation

based on adaptive dictionary learning is introduced. In Chapter 3, we run MATLAB simulations using the method discussed. Finally, we look into the future research directions in blind source separation.

2 Background

2.1 Instantaneous linear mixture model

Based on the cocktail party problem introduced in Chapter 1, here we extent the idea of blind source separation to a formal mathematical definition. The mixing process of the sources in BSS involves many models such as the instantaneous linear mixture model, the nonlinear mixture model and the convolved mixture model [24]. The instantaneous linear model omits the time delay of source propagation of reaching different observers. We assume that the instantaneous linear BSS model is adopted throughout this report.

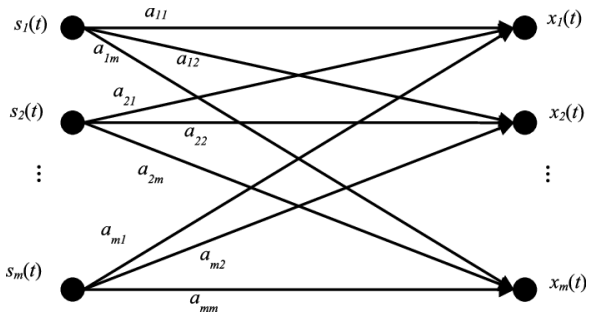


Figure 1: Instantaneous linear mixture model (m=n)

The instantaneously linear mixture model states that given m observations $\{x_1, \cdots, x_m\}$ where each $\{x_i\}_{i=1,\cdots,m}$ is a row vector of size t. Each observation is the linear mixture of n sources $\{s_1^T, \cdots, s_n^T\}$ weighted by a_{ij}

$$\forall i \in \{1, \cdots, m\}, \quad x_i = \sum_{j=1}^{n} a_{ij} s_j \tag{3}$$

Figure (1) illustrates the case when we have equal number of sources and observations. Since results are not affected by reciprocal rescaling of a_{ij} and s_j. Without loss of generality, the a_{ij} will hitherto assumed to be normalised to unit length. The mixing model can be conveniently rewritten in matrix form

$$\mathbf{X} = \mathbf{AS} + \mathbf{N} \tag{4}$$

where **X** is the observation matrix with dimension $m \times t$, **S** is the $n \times t$ source matrix and **A** is the $m \times n$ mixing matrix. An $m \times t$ matrix **N** accounts for additive noise or model imperfections. Under the blind separation problem setting, both **A** and **S** are unknown. Source separation techniques aim at recovering the original signal $\mathbf{S} = [s_1^T, \cdots, s_n^T]$ from m different mixtures by taking advantage of some information in the way the signals are mixed in observed data. In other words, source separation simply boils down to devising quantitative measures of diversity or contrast to differentiate between the sources.

Mathematically, We aim to find a demixing matrix **W** with dimension $n \times m$ which gives a linear combination of columns in the observation **X**, omitting the noise for now, that is

$$\mathbf{Y} = \mathbf{WX} \tag{5}$$

Y is hence an estimation of source **S**. Combining Eq. (4) and (5) y can be written as

$$\mathbf{Y} = \mathbf{W^T X} = \mathbf{WAS} = \mathbf{Z^T S} \tag{6}$$

Where $\mathbf{Z^T} = \mathbf{WA}$. The estimation matrix **Y** can also be represented as

$$\mathbf{Y} = \mathbf{PDS} \tag{7}$$

where **D** is a diagonal matrix and **P** is a permutation matrix that reduces the scale permutation indeterminacy of the mixing model.

2.2 Ambiguities of BSS process

Because lack of prior knowledge about the sources and mixing process. While estimating the mixing and source matrices, we introduce the diagonal matrix **D** and permutation matrix **P** in Equation (7), which accounts for the two ambiguities, amplitude and order.

Variances (energies) of each source component is unknown. The reason is that, both **S** and **A** are not defined, any scalar multiplication in one of the sources s_i would always be cancelled by dividing the corresponding column a_i by the same quantity. However, the information of sources is stored in the signal waveform rather than amplitude. This ambiguity is hence insignificant in most applications [15]. As mentioned in the last section, the columns of **A** is normalised to unity just for convenience in calculation.

Apart from amplitude uncertainty, we cannot determine the order of the sources. In Equation (3), we can freely rearrange the sources up to any permutation without affecting the observation samples. Any of the source components can be regarded as the first one. Fortunately, we can use certain techniques (e.g. Hungarian Algorithm) to sort the recovered sources after separation has been done. Again, because the information of each component is contained in the shape of waveform of that component, not the order of components. So this ambiguity is also insignificant.

2.3 Preprocessing of BSS techniques

Before applying the BSS methods on the data, it is usually very usful to do some preprocessing. In this section, we introduce two preprocessing techniques that make the blind source separation problem simpler and better conditioned. 1. Centering: Most BSS methods assumes the data to be zero centered. The most basic and necessary preprocessing is to center the observation data x_i by substract the sample mean from it. 2. Whitening: Before applying BSS methods (and after centering), the observed vector x is linearly transformed to $\tilde{x}$ so that each column is uncorrelated and have unity variance. More specifically, the covariance matrix of $\tilde{x}$ equals the identity martix, $\mathbb{E}\{\tilde{x}\tilde{x}^T\} = \mathbf{I}$. The most common used whitening transformation is eigenvalues decompostion (EVD) of the data covariance matrix.

2.4 BSS performance measures

1. Correlation coefficient measures the similarity between a recovered source $S^{'}$ and the original source S. Larger correlation coefficients indicate the original sources are better recovered.

$$\rho = \frac{\text{cov}(S^{'}, S)}{\sigma_x \sigma_y} \tag{8}$$

2. Mixing Matrix Criterion assesses the separation quality due to demixing matrix $\mathbf{A}$, especially in noisy content.

$$C_A = \|\mathbf{I_n} - P\tilde{\mathbf{A}}^{+}\mathbf{A}\| \tag{9}$$

where $\mathbf{I}$ is the identity matrix, $\mathbf{P}$ is the permutation matrix, and $\tilde{\mathbf{A}}^{+}$ is the pseudo-inverse of the estimated mixing matrix. The mixing matrix criterion is strictly positive, unless the mixing matrix is correctly estimated up to scale and permutation [1]. Low values of C_A then indicate better separation performance.

3. Human visual system (HVS): Most of out work in this report focus on blind image separation. Human visual system says that people are not as sensitive to high frequency detail as to low frequency ones. Therefore, standard metrics may not best describe the actual experiment outcomes. In order to better evaluate the results in image processing, we adopt HVS as the subjective metric.

2.5 Applications of BSS

The classical application of BSS on the cocktail party problem is trying to understand how the humans select the voice of a particular speaker from an ensemble of different voices corrupted by music and noise in the background. Other applications, besides the cocktail party problem mentioned in the introduction, have also attracted researchers' attention in the past decade. Examples are given as below.

An electroencephalogram (EEG) is a test used to find problems related to electrical activity of the brain. In EEG analysis, different artifacts such as eye-blinking deteriorate its quality. Identification of the various sources from the independent components is thus integral for clinical analysis. An innovative method combining the use of standard BSS techniques and Support Vector Machines (SVM) was applied to solve this problem [11].

Filling 'holes' in images is an interested and important inverse problem with applications in repairing the old and deteriorated artwork. Based on Morphological Component Analysis, an inpainting algorithm has been proposed which is capable of filling holes in either texture or cartoon content [12]. In Figure (2), the inpainting algorithm is applied on the famous Barbara images and achieve satisfactory result even when there are 80% missing pixels.

Figure 2: Barbara image with 80% missing pixels (right). The result of the MCA impainting is given on the left.

In Code-Division Multiple Access (CDMA), blind separation techniques are used to suppress

unintentional multiuser interference (jammer), separate the desired user signals from other users' signals [19]. BSS also has wide applications in military based telecomm systems which recovers radar reflection from strong intentional interference.

BSS is also closely related with studying the underlying factors of the financial data and driving mechanisms behind financial time series. In [18] ICA is applied to financial time series data. The data is parallel, representing the simultaneous cash flow at several stores belonging to the same retail chain. The ICA finds the fundamental factors that are common to all stores that affect the cashflow data, although each store responds to these factors in a slightly different manner. Thus, the cashflow effect of the factors specific to any particular store could be revealed.

2.6 Independent component analysis

ICA algorithms are about devising appropriate independence approximations. This includes, maximisation of non-Gaussianity, minimisation of mutual information and maximum likelihood estimation [15]. We are not going to discuss all these approximations in depth as we want to focus more on the sparsity based blind source separation. Only the FastICA and JADE are introduced below.

FastICA calculates the negentropy as an approximate of independence measure. With a taste of the central limit theorem, intuition tells us that the distribution of a sum of independent random variables tends to toward a Gaussian distribution, under certain conditions. Known from Equation (6) that $y = w^T x = wAs = z^T s$. It is clear that the closest estimation is when $z^T s = s$ which also has the least Gaussianity. Hence finding independent s is equivalent to minimisation of Gaussianity. The approximation of non-Gaussianity is based on a maximum negentropy principle.

$$J(y) \propto [E\{G(y)\} - E\{G(v)\}]^2 \tag{10}$$

Where v is a standardised Guassian variable. And G are predefined functions with the possible choices stated in [15]. In general, FastICA is based on a fixed point iteration scheme for finding the maximum of non-Gaussianity of $\mathbf{w^T x}$ in Equation (5). The basic form of FastICA algorithm is as follows.
JADE is based on similar ideas of the FastICA agorithm apart from it calculates the fourth-order statistics (Kurtosis). JADE finds out the direction where the kurtosis of observed signal grows most strongly (super-Gaussian signals) or decreases most strongly (sub-Gaussian sig-

Algorithm 1 The basic FastICA algorithm for estimating one independent component

Input:
The observed matrix $\mathbf{x}$
Output:
Estimation of $\hat{\mathbf{A}}$
1. Choose an initial (e.g. random) wright vector $\mathbf{w}$
2. Let $\mathbf{w}^+ = E\{\mathbf{x}g(w^T x) - E(g'(w^T x)\}\mathbf{w})$
3. Calculate $\mathbf{w} = \mathbf{w}^+/\|w^+\|$
4. If not converged (old and new values of $\mathbf{w}$ point in the same direction up to multiplicative signs), return to step 2

nals).

ICA suffers from several limitations which make it unsuitable in some specific applications. Firstly, ICA require the mixing matrix $\mathbf{A}$ to be full rank and square. As we mentioned in the introdcution, generally ICA cannot be applied to the underdetermined mixing scenario. Secondly, ICA also assumes that amongst the components in $\mathbf{S}$, there exists at most one component that is Gaussian. This means ICA is not robust under the additive Gaussian nosie setting. While even implicit, the ICA algorithm requires information on the source distribution when doing separating computation such as maximum likelihood estimation, making it hard for model generalisation. We will exam the limitations of ICA in future simulations.

2.7 Sparse representations of signals

Here we introduces the idea of sparse signal processing and how it helps to solve underdetermined linear systems (dictionary). First we need to articulate the expression from the underdetermined system mentioned in previous parts. In BSS, 'underdetermined system' means linear combination of **source signals** whereas here we refer to further linearly decompose the source signals into a given **dictionary**. In a more plain language, less number of examples than the data dimensionality involved are available for learning the dictionary. Sparse signal processing has numerous applications in compress sensing, image deniosing and super-resulation reconstrcution.

2.7.1 Overcomplete dictionary

The key idea of sparse signal representation is to assume that the sources are sparse, or can be decomposed into the combination of a small number of signal components. By sparse, we mean that most values in the signal or its transformed coefficients are zero. These signal

components are called atoms, and the collection of all the atoms is referred to as a dictionary [17]. In the general sparse representation framework, we can model a signal $y \in R^N$ as the linear combination of D elementary signal atoms in dictionary $\mathbf{\Phi}$, such that.

$$y = \alpha\mathbf{\Phi} \tag{11}$$

where α is called the representation coefficients of y in the dictionary $\mathbf{\Phi}$ (the $N \times D$ matrix). In the case of overcomplete representations, the number of waveforms or atoms (φ_i) is higher than the dimension of the space in which y lies, that is $D > N$.

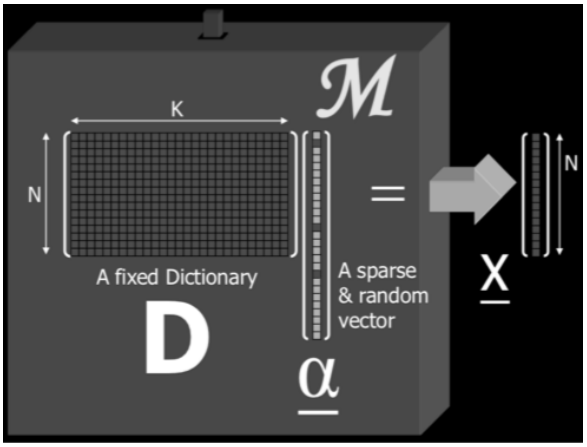


Figure 3: Illustration of sparse representation using overcomplete dictionary

2.7.2 Sparse decomposition

The decomposition problem of a signal or image in predefined $\mathbf{\Phi}$ amounts to recovering the coefficient vector α in Equation (11). When $\mathbf{\Phi}$ is overcomplete, the solution is generally not unique. In that case, our goal is to recover the sparsest solution α which requires solving:

$$\min_{\alpha} \|\alpha\|_0 \quad \text{s.t.} \quad y = \alpha\mathbf{\Phi} \tag{12}$$

However, the above equation leads to an NP-heard optimisation problem due to its non-convextivity. Alternatively, we convexify the constraint by substituting the convex ℓ_1-norm with the ℓ_0-norm, leading to the following equation:

$$\min_{\alpha} \|\alpha\|_1 \quad \text{s.t.} \quad y = \alpha\mathbf{\Phi} \tag{13}$$

or

$$\min_{\alpha} \|\alpha\|_1 \quad \text{s.t.} \quad \|y - \alpha\mathbf{\Phi}\|_2 \leq \sigma \tag{14}$$

Such relaxations have led to a wide range of algorithms for signal reconstruction: the *Basis Pursuit* based on linear programming [8], the greedy algorithm such as *Matching Pursuit* [17] and subspace techniques such as *Subspace Pursuit* [9] and *CoSaMP* (Compressive Sampling Matching Pursuit) [3]. In the next section, we provide essential insights into the use of sparsity in BSS and we highlight the central role played by morphological diversity as a way of contrast between the sources.

2.8 Morphological component analysis (MCA)

We now introduce a practical algorithm named the morphological component analysis (MCA) aiming at decomposing signals in overcomplete dictionaries composed of a union of orthonormal basis, i.e. our dictionary is a concatenation of sub-dictionaries. In MCA setting, y is a linear combination of D morphological components.

$$y = \sum_{i=1}^{D} \alpha_i \mathbf{\Phi}_i = \sum_{i=1}^{D} \varphi_i \tag{15}$$

where $\{\mathbf{\Phi}_i\}$ are orthonormal basis whose columns are the atoms and is general normalised to a unit ℓ_2-norm. Morphological diversity then relies on the incoherence between the subdictionaries. In terms of ℓ_0-norm, this morphological diversity can be formulated as follows:

$$\forall\{i,j\} \in \{1,\cdots,D\}; \quad j \neq i \Rightarrow \|\varphi_i \Phi_i^T\|_0 < \|\varphi_j \Phi_i^T\|_0 \tag{16}$$

Intuitively, we can always find a sub-dictionary that one morphological component is highly sparse in it whereas other components are not very sparse. We therefore estimate the morphological components $\{\varphi_i\}_{i=1,\cdots,D}$ by solving the following convex minimisation problem.

$$\{\varphi_i\} = \text{Arg} \min_{\{\varphi_i\}} \sum_{i=1}^{D} \|\varphi_i \mathbf{\Phi}_i^T\|_1 + k \|y - \sum_{i=1}^{D} \varphi_i\|_2^2 + \sum_{i=1}^{D} \gamma_i C_i(\varphi_i) \tag{17}$$

where C_i implements constraints (e.g. TV correction) on component φ_i. Note that this minimisation problem differs from the problem setting in Equation (13) by relaxing the equality constraints to the later punishment term. A fast numerical solver called the *Block-Coordinate Relaxation Method* [21] was proposed to solve this kind of optimisation problem. The algo-

rithm is given as follows:

Algorithm 2 The numerical algorithm for MCA

Input:
The sources y, dictionary Φ, number of morphological components D, number of iterations L_{max} and threshold δ

Output:
Each morphological components S_k
Initialize L_{max}; number of iterations; threshold $\delta = kL_{max}$;
1. Perform J times:
    2. Perform D times:
    Update of φ_i assuming all φ_l, $l \neq i$ are fixed:
    - Calculate the residual $r = \varphi \sum_{l=1, l\neq i}^{D} \varphi_l$
    - Calculate the transform $\mathbf{\Phi^T}$ of $\varphi_i + r$and obtain $a_i = \mathbf{\Phi_i^T}(\varphi_i + r)$
    - Soft threshold the coefficient a_i with the δ threshold and obtain $\hat{a}_i$.
    - Reconstruct φ_i by $\varphi_i = \mathbf{\Phi}_i \hat{a}_i$
    - Apply the constraint correction $\varphi_i = \varphi_i - \mu\gamma_k \frac{\partial C_i}{\partial s_i}$
    - The parameter μ is chosen either by a line-search minimizing the overall
3. Update the threshold by $\delta = \delta - k$.
4. If $\delta > k$, return to Step 2. Else, finish.

2.9 Mutichannel morphological component analysis (MMCA)

In this section, we extend the monochannel sparse decomposition problem described and characterized in last section to multichannel data. In the MMCA setting, we assumed that the sources $\mathbf{S}$ in Equation (4) have strictly different morphologies (i.e. each source s_i is assumed to be strictly sparsely represented in one particular orthonormal basis $\mathbf{\Phi}_i$) [5]. Figure 4 precisely reveals the difference between MCA and MMCA. The top observation can be strictly divided in to texture and gaussian parts whereas the bottom multichannel observations are complex combinations of curvelets and texure components.

An iterative thresholding Block-Coordinate Relaxation algorithm similar to Algorithm (2) was proposed to solve the joint optimisation problem below.

$$\{\tilde{\mathbf{A}}, \tilde{\mathbf{S}}\} = \text{Arg} \min_{\mathbf{A},\mathbf{S}} \sum_{k=1}^{N} \|s_k \mathbf{\Phi}_k^T\|_1 + \lambda \|\mathbf{X} - \mathbf{AS}\|_2^2 \tag{18}$$

The equation above is very similar to Equation (17) in MCA. Unfortunately, this MMCA criterion suffers from several drawbacks and particularly from an indeterminacy attached to the

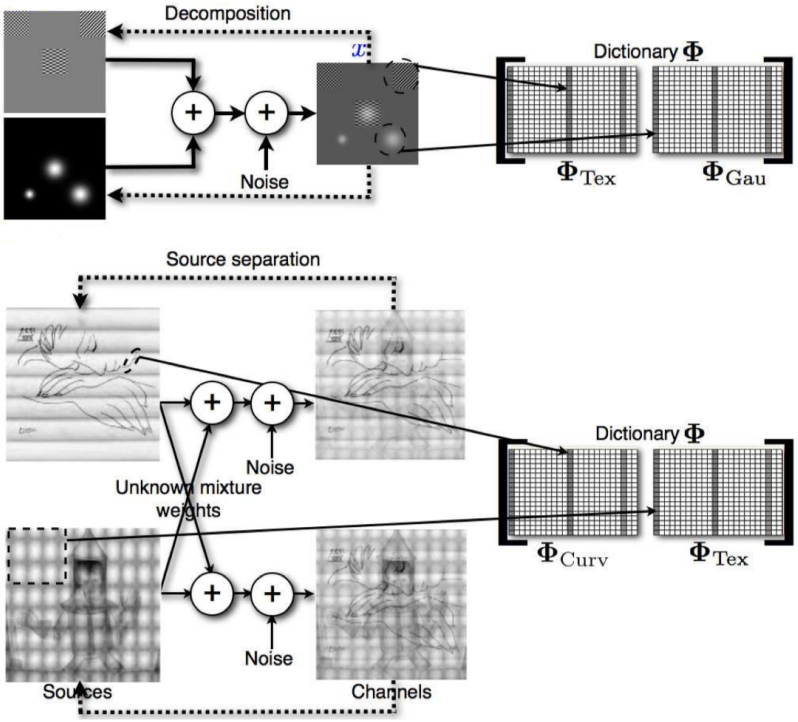


Figure 4: Illustration of the image decomposition(top) using MCA and the blind source separation (bottom) using MMCA. For the bottom part, each source is itself a mixture of morphological components(texture and cartoon)

model structure [5]. For instance, the minimisation of this criterion may result in trivial solutions : $\mathbf{A} = \rho\mathbf{A}$ and $\mathbf{S} = \frac{1}{\rho}\mathbf{S}$ (the sparsity term can be minimised as desired as long as ρ tends to $+\infty$) [5]. We therefore normalise the columns in matrix $\mathbf{A}$ at each iteration, as mentioned in Chapter 2.2.

Let's introduce the k^{th} channel residual $D_k = X - \sum_{j \neq k} a^j s_j$ (accounts for the part of that data unexplained by the other couples $\{a^j, s_j\} j \neq k$), the minimisation of the whole criterion (18) is equivalent to this joint minimisation problem.

$$\{\tilde{s}_k, \tilde{a}^k\} = \text{Arg} \min_{\mathbf{A},\mathbf{S}} \|s_k \mathbf{\Phi}_k^T\|_1 + \lambda \|D_k - a^k s_k\|_2^2 \tag{19}$$

If we also assume the noise covariance matrix Γ_n is known, the criterion new becomes:

$$\{\tilde{\mathbf{A}}, \tilde{\mathbf{S}}\} = \text{Arg} \min_{\mathbf{A},\mathbf{S}} \|s_k \mathbf{\Phi}_k^T\|_1 + Trace\{(D_k - a^k s_k)\Gamma^{-1}(D_k - a^k s_k)^T\} \tag{20}$$

Zero the garident with respect to s_k and a^k leads to the following coupled equations

$$\begin{cases} s_k = \frac{1}{a^k \Gamma^{-1} a^k}({a^k}^T - \frac{1}{2\lambda_k} Sign(s_k \mathbf{\Phi}_k)\mathbf{\Phi}_k^T) \\ a^k = \frac{1}{s_k s_k^T} D_k s_k^T \end{cases} \tag{21}$$

We then use the soft thresholding algorithm to solve for approximation of s_k and a_k. Setting the threshold $\delta = \frac{\lambda_k}{2\|a^k \Gamma^{-1} a^k\|}$. Then considering a fixed s_k, the update on a_k follows a simple least square linear regression.

Algorithm 3 The numerical algorithm for MMCA

Input:

The sources S, dictionary Φ, number of morphological components N, number of iterations L_{max} and threshold $\delta = k \cdot L_{max}$

Output:

Estimation $\hat{\mathbf{A}}$ and $\hat{\mathbf{S}}$

1. Perform J times:
    2. Perform N times:
        - Normalisation and propagation of a^k, s_k, δ_k for scale invariance:
        - Estimation of s_k assuming all s_l , $l \neq k$ and a_l are fixed
        - Calculate the residual $D_k = X - \sum_{l=1, l\neq k} a^l s_l$
        - Projection the residual $\hat{s}_k = \frac{1}{{a^k}^T \Gamma_n^{-1} a^k} \Gamma_n^{-1} D_k$
        - Calculate $a_k = \hat{s}_k {\mathbf{\Phi}^{\mathbf{T}}}_k$
        - Soft-thresholding the coefficients a_k with the δ_k threshold and obtain $\hat{a}_k$
        - Reconstruct s_k by $s_k = \hat{a}_k \mathbf{\Phi}_k$
        - Estimation of a_k by a_k assuming all s_l and $a^l_{l\neq k}$ are fixed $a_k = \frac{1}{s_k s_k^T} D_k s_k^T$
3. Update the threshold by $\delta = \delta - k$.
4. If $\delta > k$, return to Step 2. Else, finish.

2.10 Generalised morphological component analysis (GMCA)

We now apply the idea of morphological diversity to more generalised blind source separation problems. In GMCA, we assume each source is modelled as a weighted sum of D morphological components where each component is sparsely represented in a specific basis. GMCA pursuits an unmixing scheme, through the estimation of $\mathbf{A}$, which leads to the sparsest sources $\mathbf{S}$ in the dictionary $\mathcal{D}$. This is expressed in a Lagrangian form:

$$\{\tilde{\mathbf{A}}, \tilde{\mathbf{S}}\} = \text{Arg} \min_{\{\mathbf{A},\mathbf{S}\}} \sum_{i=1}^{n} \sum_{k=1}^{D} \|\varphi_{ik} \mathbf{\Phi}_k^T\|_1 + \lambda \|\mathbf{X} - \mathbf{AS}\|_2^2 \tag{22}$$

The product **AS** can be split into $n \times D$ multichannel morphological components: $AS = \sum_{i,k} a^i \varphi_i k$. Based on this decomposition, an alternating minimisation algorithm was proposed to estimate iteratively one term at a time [5]. Again, each column of **A** is forced to have unit norm at each iteration to avoid the classical scale indeterminacy of the product in (22). Define the multichannel residual by $\mathbf{X}_{i,k} = \mathbf{X} - \sum_{\{p,q\}\neq\{i,k\}} \alpha^p \varphi_{pq}$ as part of the data unexplained by the multichannel morphological component $\alpha^i \varphi_{ik}$. Estimating the morphological component $\varphi_{ik} = \alpha_{ik}\mathbf{\Phi}_k$ assuming **A** and $\varphi_{\{pq\}\neq\{ik\}}$ are fixed leads to:

$$\tilde{\varphi}_{ik} = \arg\min_{\{\varphi_{ik}\}} \|\varphi_{ik}\mathbf{\Phi}_k^T\|_1 + \lambda \|\mathbf{X}_{i,k} - a^i \varphi_{ik}\|_2^2 \tag{23}$$

Similarly to MMCA, GMCA uses the component-wise iterative thresholding algorithm which is summarized as follows. Note the difference notations of morphological components coefficients α_{ik} and mixing matrix coefficients a^i.

Algorithm 4 The numerical algorithm for GMCA.

Input:

The sources S, dictionary Φ, number of morphological components D, number of iterations L_{max} and threshold $\delta = k \cdot L_{max}$

Output:

Estimation $\tilde{\mathbf{A}}$ and $\tilde{\mathbf{S}}$

1. Perform n times:
    2. Perform D times:
    - Compute the residual term r_{ik} assuming the current estimates $\tilde{\varphi}_{\{pq\}\neq ik}$ are fixed; $r_{ik} = \tilde{a}^{i\,T}(\mathbf{X} - \sum_{\{p,q\}\neq\{i,k\}} \tilde{a}^p \tilde{\varphi}_{pq})$
    - Estimate the current coefficients of $\tilde{\varphi}_{ik}$ by thresholding with threshold δ $\tilde{\alpha}_{ik} = \lambda_\delta(r_{ik}\mathbf{\Phi}_k^T)$
    - Reconstruct φ_{ik} by $\varphi_{ik} = \tilde{\alpha}_{ik}\mathbf{\Phi}_k$
    - Estimation of a_k by assuming all φ_{pq} and $a^{p\neq k}$ are fixed $\tilde{a}^i = \frac{1}{\tilde{s}_i^2}(\mathbf{X} - \sum_{p\neq i}^n \tilde{a}^p \tilde{s}_p)\tilde{s}_i^T$
3. Update the threshold by $\delta = \delta - \lambda$.
4. If $\delta > k$, return to Step 2. Else, finish.

2.11 FastGMCA algorithm (FGMCA)

In the last section, we described GMCA algorithm which needs the projection of the residual into the dictionary space at each iteration $\tilde{\alpha}_{ik} = \lambda_\delta(r_{ik}\mathbf{\Phi}_k^T)$. Note that the application of $\mathbf{\Phi}_k^T$ will consume most of the computation power. Thus, GMCA could be very computationally

demanding for large scale, high dimensional problems [5]. In practice, we apply the an improved version coined fast GMCA by adding some assumptions to the original problem.

We assume each row of $\mathbf{\Theta_X} = \mathbf{X}D^T$ stores the decomposition of each observed channels in D. And each row of $\mathbf{\Theta_S} = \mathbf{S}D^T$ stores the decomposition of each source. If the supports (ℓ_0 decompostion) of $\mathbf{X}$ and $\mathbf{S}$ satisfies

$$\Delta_D(x_i) = \sum_{j=1}^{n} \alpha_{ij} \Delta(s_j) \tag{24}$$

Then we can rewrite the Lagrangian form as follows.

$$\{\tilde{\mathbf{A}}, \tilde{\mathbf{S}}\} = \text{Arg} \, \|\mathbf{\Theta_S}\|_0 + \lambda \, \|\mathbf{\Theta_X} - \mathbf{A}\mathbf{\Theta_S}\|_2^2 \tag{25}$$

To conclude, the fast GMCA algorithm works in the sparse transformed domain and omits the dictionary decomposition process at each iteration. Thus we can precompute the projection of the mixtures in, for example the wavelet domain and run the FastGMCA algorithm in that domain. This will significantly accelerate our blind separation process. Now write fast GMCA in a stepwise flavour.

Algorithm 5 The numerical algorithm for FastGMCA

Input:

The obervations $\mathbf{Y}$, dictionary $\mathbf{\Phi}$, number of morphological components N, number of iterations L_{max} and threshold $\delta^{(0)} = k \cdot L_{max}$

Output:

Estimation $\tilde{\mathbf{A}}$ and $\tilde{\mathbf{S}}$

1. Decompose the mixture in transformed domiain an obtain $\mathbf{\Theta_X}$
2. While each δ is higher than a given lower threshold $\delta^{(0)}$:
   - Update $\mathbf{\Theta_S}$ with thresholding operator λ_δ and $\mathbf{A}$ is fixed at h^{th} iteration.
     $\hat{\mathbf{\Theta}}_S^{(h+1)} = \lambda_\delta(A^{\dagger(h)}\mathbf{\Theta_X})$
   - Update $\mathbf{A}$ by a least-square estimate assuming $\mathbf{\Theta_S}$ is fixed.
     $\hat{\mathbf{A}}^{(h+1)} = \mathbf{\Theta_X}\hat{\mathbf{\Theta}}_{\mathbf{S}}^{(h)^T}(\hat{\mathbf{\Theta}}_{\mathbf{S}}{}^{(h)}\hat{\mathbf{\Theta}}_{\mathbf{S}}^{(h)^T})^{-1}$
   - Decrease δ.

2.12 Blind source separation based on adaptive dictionary learning

In previous sessions, we mentioned about sparse decomposition using prescribed overcomplete dictionaries such as Wavelet, Curvelet or unions of orthonormal transform bases. This method works well when the original sources have components that are largely different

from each other in the transform domain. However, this may not lead to the most sparsified decomposition of each individual sources, or in another word, the dictionaries found may not be appropriate in the sense that they may fit better the mixtures rather than the sources.

As described before, dictionary learning has wide applications in compress sensing and image denoising. Apart from the pursuit algorithms described in 2.7.1 that finds the sparse coefficients with respect to a given dictionary. Recent research is concentrated on obtaining a adapting dictionary in order to achieve best sparse signal representations. Michal Aharon designed the K-SVD [2] algorithm which uses a K-means clustering like algorithm and columnwise updating flavor of the dictionary atoms. Dai and Tao proposed the SimCo method [10] based on K-SVD which avoids falling into a singular point during the optimisation process. It is worthy noting that these methods only gives an appropriate solution to the dictionary learning problem formulated in in Equ.(26), but finding a global minima (for sure) is still an open question.

$$\min_{\alpha} \|\alpha\|_1 \quad \text{s.t.} \quad \|\mathbf{Y} - \mathbf{X}\mathbf{\Phi}\|^2 \leq \sigma \tag{26}$$

Motivated by the idea of image denoising we can adapt MMCA to learned local dictionaries from the mixed sources within the separation process[1]. Hence we does not need any prior knowledge about the sparse domain of the sources. We now introduce the K-SVD dictionary learning algorithm which will be applied later in experiment section.

2.12.1 K-SVD dictionary learning

The K-SVD algorithm can be decomposed into two stages, which are executed alternatively as what we did in the iterative relaxation method. First we can use any pursuit algorithm (BP, OMP) to calculate the sparse coefficients $\mathbf{X}$. This is called the sparse coding stage. Then we proceed to the codeword update stage. In Equation (26), assuming that both $\mathbf{\Phi}$ and $\mathbf{X}$ is fixed and we only consider one atom ϕ_k in dictionary $\mathbf{\Phi}$ and the coefficients x^k (k^{th} row in $\mathbf{X}$) corresponding to it, the penalty term in the objective function can be rewritten as

$$
\begin{aligned}
\|\mathbf{Y} - \mathbf{\Phi X}\|^2 &= \|\mathbf{Y} - \sum_{j=1}^{K} \phi_j x_T^j\|^2 \\
&= \|(\mathbf{Y} - \sum_{j \neq k} \phi_j x_T^j) - \phi_k x_T^k\|^2 \\
&= \|\mathbf{E}_k - d_k x_T^k\|^2
\end{aligned}
\tag{27}
$$

where $\mathbf{E}_k$ stands for the residual. In order to force the sparsity in $\mathbf{X}$, we need to extract the columns in $\mathbf{E}_k$ correspond to non-zero elements in x^k. That is,

$$
\begin{aligned}
w_k &= \{i | 1 \geq i \leq K, x^k(i) \neq 0\} \\
\Omega_k &= \text{concatenation of N} \times \text{w}_k; \\
\mathbf{E}_k &= \Omega_k \mathbf{E}_k
\end{aligned}
\tag{28}
$$

Ω_k is a mask consisting 0 and 1s so that we only consider atoms that refer to a certain row in the sparse coefficients. Hence we have split the term $\mathbf{\Phi X}$ to k rank-1 matrices. Among those, only the k^{th} atom remains in the question. This is a least square problem that can be directly solved using singular value decomposition (SVD).

$$
\mathbf{E}_k = U \Sigma V^T \tag{29}
$$

We define the solution for ϕ_k as the first column of U and the coefficient vector x^k as the first column of V. In the same manner, K-SVD sweeps through all columns always use the most updated coefficients as they emerge from preceding SVD steps. To conclude, The K-SVD algorithm obtains the dictionary update by K separate SVD computations, which explains its name.

2.12.2 K-SVD+MMCA

The combination of K-SVD and MMCA in blind source separation is similar to the conventional MMCA algorithm apart from it requires updating of three matrices, the estimated mixing $\mathbf{A}$, the estimated source $\mathbf{S}$ and the dictionary $\mathbf{\Phi}$. A stepwise algorithm is displayed below

Algorithm 6 The numerical algorithm for K-SVD+MMCA

Input:

The sources S, dictionary Φ, number of morphological components N, number of iterations L_{max} and threshold $\delta^{(0)} = k \cdot L_{max}$.

Output:

Estimation $\hat{\mathbf{A}}$ and $\hat{\mathbf{S}}$.

1. Initialse $\mathbf{\Phi}$ to a known overcomplete dictionary.
2. Set $\mathbf{A}$ to a radom column-normalised matrix.
3. $\mathbf{X} = A^T Y$.
4. for L_{max} iterations:
   for $j = 1 : N$:
   - Extract patches from x_j.
   - Update coefficient a_j using OMP.
   - Update Φ_j using K-SVD.
   - Calculate the residual $\mathbf{E_j} = \mathbf{Y} - \sum_{l \neq j} a_l x_l^T$.
   - Compute x_j.
   - $a_j = \mathbf{E_j} x_j$
   - Normalise a_j
5. Decrease σ until stopping criterion is met.

3 Block Sparse K-SVD algorithm applied to BSS

Last section proves the strength of sparsity based methods applied to blind source separation problem. Inspired by using adaptively learned local dictionary in MMCA/GMCA, we aim to create a new adaptive dictionary learning algorithm combined with BSS. The idea driven behind is simple that, certainly will we acquire a better separation result if we improve the level of sparsity of the dictionary,.

A variety of sparse dictionary learnging algorithm have been proposed in the literature for this purpose, based on the K-SVD algorithm. The block-sparse dictionary learning algorithm is proposed by Lihi in [25]. It exploits the hidden structure that is intrinsic in the signals for producing more efficient sparse representations. In the following content, we introduce the theory of this algorithm and try to embed it in the blind source separation process. The algorithm consists of two steps: a block structure update step (SAC) and a dictionary update step (BK-SVD).

3.1 Problem definition

Given a set of signals $\mathbf{Y} = \{y_i\} \in R^N$, we wish to find an overcomplete dictionary $\mathbf{\Phi}$ in $R^{N\times K}$ whose atoms are sorted in blocks, correspondingly the non-zero coefficients representations $\mathbf{X} = \{x_i\}$ are concentrated in a fixed number of blocks. More specifically, suppose dictionary atoms sorted in blocks that enable **block-sparse** representations of input signals. Each block has its own label, indexed as d_i. We claim that a vector $\mathbf{X} \in R^K$ is k-block-sparse over d if its non-zero entries under certain block strucutre is less than k. This is denoted by

$$\|x\|_{0,d} = k \tag{30}$$

which means there are k number of non-zero blocks having block structure d. Figure 5 presents two equivalent examples, The dictionary can be expressed as 5 blocks but with 2-block-sparse presentations.

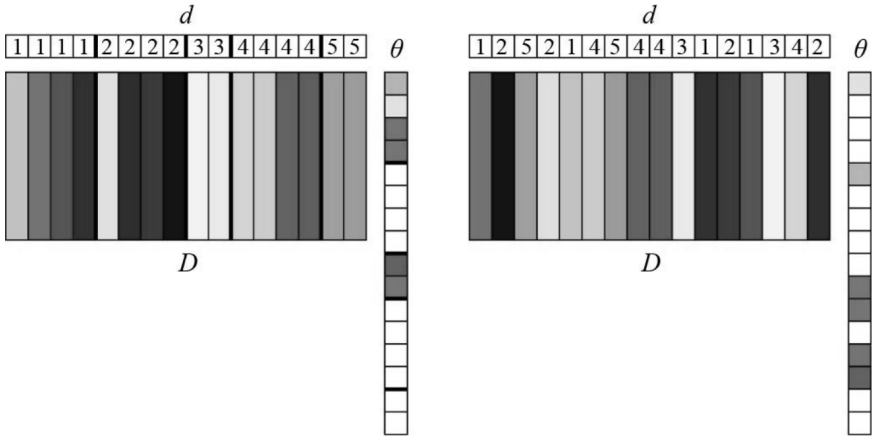


Figure 5: Two equivalent representations with different block structure.

We can now formulate the problem is a mathematical way that we intent to find a dictionary $\mathbf{\Phi}$ and a block structure d, with maximal block size s, that lead to optimal k-block sparse representations $\mathbf{X} = \{x_i\}_i^L$ for signals in $\mathbf{Y}$. The objective function is given as below.

$$\begin{aligned} &\min_{D,d,\mathbf{X}} \|\mathbf{Y} - \mathbf{\Phi}\mathbf{X}\| \\ \text{s.t.} \quad &\|x_i\|_{0,d} \leq k,\ i = 1,...L \\ &|d_j| \leq s,\ j = 1,....B \end{aligned} \tag{31}$$

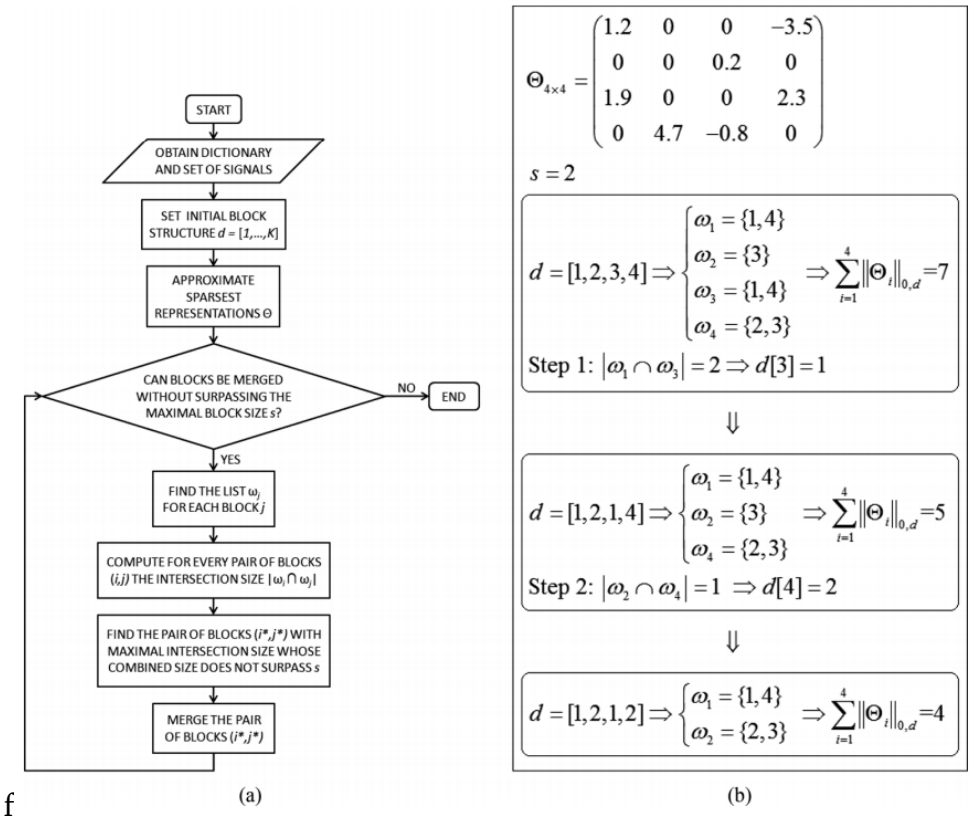
where d_j is the set of indices belonging to block j, s is the maximum block size and B is is the total number of blocks (number of dictionary columns divided by the block size). When the

maximal block size is set to 1, the proposed algorithm reduces to normal K-SVD.

3.2 Algorithm preview

Like other optimisation problems in previous sections, the problem in Equation (31) is non-convex. We therefore adopt the coordinate relaxation technique. The initial dictionary can be set up as a DCT dictionary or any random collection of K signals. Then the block structure is solved by the *sparse aggolmerative clustering* (SAC) algorithm[16]. Agglomerative clustering is a 'bottom-up' approach who groups according to distance metric (i.e. ℓ_0 norm). SAC algorithm solves for an optimal block structure refer to input command (k and s) while keeping the dictionary fixed.

$$\begin{aligned} \left[d^{(m)}, \mathbf{\Phi}^{(m)}\right] = \arg\min_{X,d} \quad & \|\mathbf{Y} - \mathbf{\Phi}^{(m-1)}\mathbf{X}\|_2 \\ \text{s.t.} \quad & \|x_i\|_0, d \le k, \quad i = 1,...,L \\ & |d_j| \le s,\ j = 1,....B \end{aligned} \tag{32}$$



f

Figure 6: (a) A flow chart about the SAC algorithm; (b) A stepwise example of the SAC algorithm.

Figure (6) presents a detailed example of the clustering process of the SAC algorithm. In this example, the maximal block size is set to be 2. At the first iteration, row 1 and row 3 in the dictionary has the largest intersection, by which means their ℓ_0 norm has the largest similarity ($\omega = \{1, 4\}$). Consequently, they are merged together. In the next step, row 2 and row 4 have the largest intersection and are merged. Up to now, no acquisitions an be further executed because of the maximal block size is limited by 2.

One may wonder that the SAC algorithm only deals with the sparse coefficients but not the dictionary directly. This is because rows of the coefficients $\mathbf{X}$ exhibits a similar pattern on non-zeros as the columns of the dictionary block. In other words, grouping coefficients is equivalent to grouping the dictionary atoms according to the sparsity pattern. Furthermore, merging blocks is always beneficial to the reconstruction accuracy of one dictionary, due to the reconstruction result is a liner summation of atoms, grouping only rewrite the coefficients in $\mathbf{X}$, which does not deprave the final result.

After having successfully determined the optimal block structure. We then solve for new dictionary while keeping $\mathbf{\Phi}^m$ fixed. The objective function becomes.

$$\begin{aligned} \left[\mathbf{X}^{(m)}, \mathbf{\Phi}^{(m)}\right] = \arg\min_{\Phi, X} \quad & \|\mathbf{Y} - \mathbf{\Phi}^{(m-1)}\mathbf{X}\|_2 \\ \text{s.t.} \quad & \|x_i\|_0, d \leq k, \quad i = 1, ..., L \end{aligned} \tag{33}$$

The author in [25] proposed block K-SVD (BK-SVD), a natural extension of the K-SVD algorithm to solve (33). BK-SVD algorithm employs a similar 'columnwise' atom updating manner as in K-SVD but forces the learned dictionary to have higher sparsity level. We first fix $\mathbf{\Phi}^{m-1}$ and can use any pursuit method (e.g. Batch-OMP) to solve (33) which reduces to.

$$\begin{aligned} \mathbf{X}^{(m)} = \arg\min_{X} \quad & \|\mathbf{Y} - \mathbf{\Phi}\mathbf{X}^{(m-1)}\|_2 \\ \text{s.t.} \quad & \|x_i\|_0, d \leq k, \quad i = 1, ..., L \end{aligned} \tag{34}$$

Then, to obtain $\mathbf{\Phi}^{(m)}$, fix $\mathbf{X}^{(m)}$, d and $\mathbf{Y}$. The calculation procedure is same as K-SVD, where the blocks of atoms (not standalone atoms) in the dictionary is updated sequentially, alongside with the corresponding nonzero coefficients in $\mathbf{X}^{(m)}$. The key difference between K-SVD and BK-SVD is that, in K-SVD only the highest rank component of the residual is updated, resulting in one atom change. Conversely in BK-SVD, atoms in the same block can be updated simultaneously.

We have now introduced the whole block sparse dictionary learning algorithm. Extend the idea of K-SVD+MMCA algorithm, we can replace the step of calculating sparse coefficients and dictionary atoms by the newly proposed SAC+BK-SVD algorithm. The overall BSS framework using SAC+BK-SVD is summarised below.

Algorithm 7 The numerical algorithm for SAC+BK-SVD+MMCA

Input:

The sources S, dictionary Φ, number of morphological components N, number of iterations L_{max} and threshold $\delta^{(0)} = k \cdot L_{max}$.

Output:

Estimation $\hat{\mathbf{A}}$ and $\hat{\mathbf{S}}$.

1. Initialse $\mathbf{\Phi}$ to a known overcomplete dictionary.
2. Set $\mathbf{A}$ to a radom column-normalised matrix.
3. $\mathbf{X} = \mathbf{A}^T\mathbf{Y}$.
4. for L_{max} iterations:
   for $j = 1 : N$:
   - Extract patches from x_j.
   - Fix dictionary $\mathbf{\Phi}^{(m-1)}$ and update coefficients α^m and block structure $d^{(m)}$ by applying sparse agglomerative clustering.
   - Fix the block structure $d^{(m)}$ and update the dictionary atoms $\mathbf{\Phi}^{(m)}$ by applying BK-SVD.
   - Calculate the residual $\mathbf{E_j} = \mathbf{Y} - \sum_{l \neq j} a_l x_l^T$.
   - Compute x_j.
   - Calculate mixing matrix column $a_j = \mathbf{E_j} x_j$
   - Normalise a_j
5. Decrease σ until stopping criterion is met.

3.3 Complexity analysis

The standard algorithm for agglomerative clustering (SAC) has a time complexity of $\mathcal{O}(K^3)$ (K is the number of dictionary atoms in this occasion). This makes it not suitable for even medium datasets. However the complexity of BK-SVD requires s times less number of singular value computations than the K-SVD algorithm. Because of the simultaneous updating of atoms belong to the same block. As proved in [20], K-SVD algorithm has a complexity of $\mathcal{O}((ks)^2K + 2NK)$. Because of the block nature of BK-SVD, we expect BK-SVD to have a complexity of $\mathcal{O}((k)^2K + 2NK)$, where k is the sparsity level (number of sparse blocks) and s is the maximal block size. The total compexity of combined SAC+BK-SVD is therefore $\mathcal{O}((k)^2K + 2NK + K^3)$. Therefore the overall convergence rate of the proposed BSS framework will be slower than simple K-SVD method.

4 Experiments on Image Source Separation

4.1 Software requirements

The software in this project has prerequisite on several opensource Matlab Toolboxes (i.e. WaveLab 850, K-SVD, MCALab110 and image processing toolbox). Complete version of my code can be found on Github[1]. In addition, I wrote my own block-sparse BSS Toolbox which involves some novelty.

4.2 Solve the BSS scale and permutation indeterminacy

As mentioned in previous sections, BSS methods suffers from ambiguities that the estimated source can be scaled permuted up to any arbitrary order. The solution to this problem is equivalent to an optimal assignment algorithm. An intuitive method would be comparing the similarity metric (e.g. Euclidean distance) between every estimated source and the original sources. It is easy to see that the complexity is $O(n^2)$. Unfortunately, the optimal assignment may not, in general, be attained in this way [23]. In this report, we employ the Kuhn-Munkres Algorithm (also well-known as the Hungarian Algorithm) with a higher complexity $O(n^3)$ (n is the number of channels) but guarantees the optimal assignment to calculate the permutation matrix.

4.3 Image decomposition

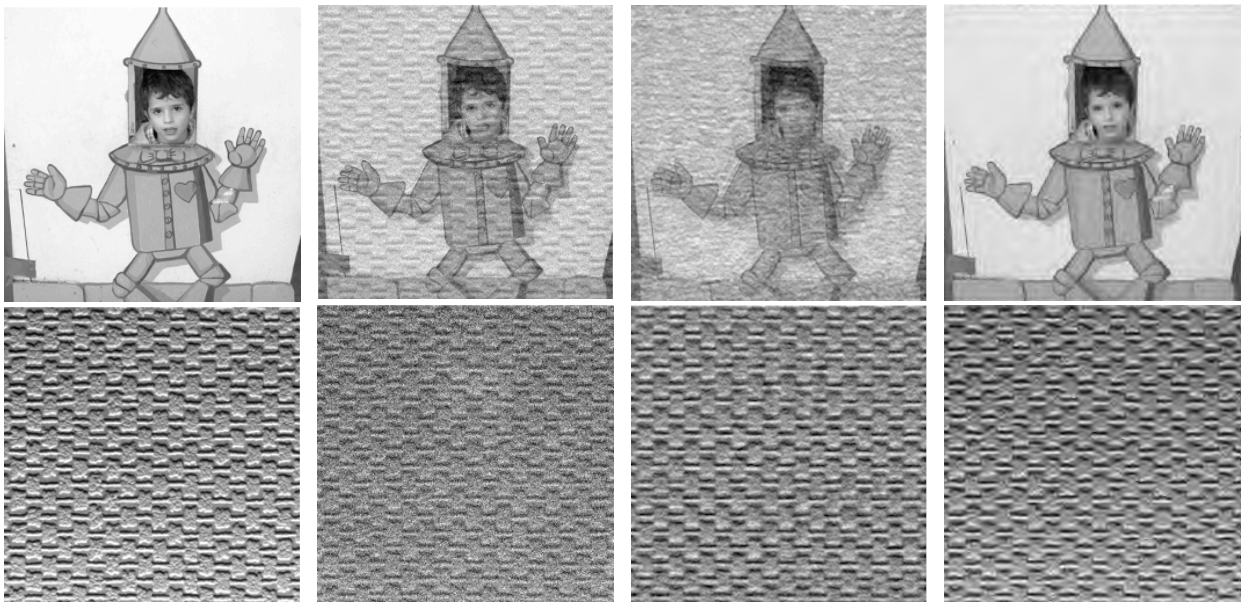
In this section, we turn to use MMCA to separate two-dimensional data and compare the result with the standard ICA source separation techniques. In Figure (7a) are two source signals, one of which is oscillating textures while another is a 'boy' image. Curvelet transform is selected as the dictionary for source 1 and discrete consine transform is selected for source 2. This is similar to the idea of a double sparse dictionary therefore we can decompose the mixtures into cartoon and texture. Figure (7c) illustrates the separated image using MCA under the presence of 20dB Gaussian noise. It can be shown that MCA is able to split the texture and cartoon parts. However, the reconstruction quality of the 'boy' image using curvelet dictionary in MCA does not give satisfactory result though. We think the decomposition presenting in the dictionary domain may not be extremely sparse, as some of the mixtures can still be seen in the output image.

[1] https://github.com/Dieselmarble/FYP

Note that the MCA algorithm, unlike BSS methods, only takes one single combination of sources ($m = 1$). Now we extend the observed mixtures to a multichannel case and BSS techniques applies. The correlation coefficient between two sources are only 0.07. Hence the independence assumption for ICA methods is valid here. We create four mixtures from two source images. Figure (7d) shows that MMCA is clearly able to efficiently separate the original source images, achieving better visual results than FastICA in (b). Quantitatively, Figure 8 shows the correlation between the original sources and those estimated. As the data noise variance increase, MMCA (dashed line) clearly achieves better estimation quality and shows clear robustness compared to non de-noised ICA methods. In addition, one can note that both JADE and FastICA provides similar performance.

Figure 8 plots the matrix estimation error is defined as $\|\mathbf{I_n} - P\tilde{\mathbf{A}}^{+}\mathbf{A}\|$, after elimination the effect of the permutation and scale indeterminacy. Contrasting with standard ICA methods, MMCA iteratively estimates the mixing matrix from coarse (i.e. smooth) versions of the sources and thus is not penalized by the presence of noise. As a consequence, MMCA is clearly more robust to noise than standard ICA methods, even under very noisy context [4]. This result reflects our expectation in section 2.6 that ICA is not robust under the additive Gaussian noise setting.

The results in this experiment proves that sparsity based methods successfully handle the image segmentation task. Especially under the multichannel case, MMCA significantly outperforms the standard ICA methods in terms of separation quality and robustness.



(a) (b) (c) (d)

Figure 7: Experiment1: Image segmentation (PSNR = 20dB); **(a)**: Original sources; **(b)**: FastICA outputs; **(c)**: MCA outputs; **(d)**: MMCA outputs

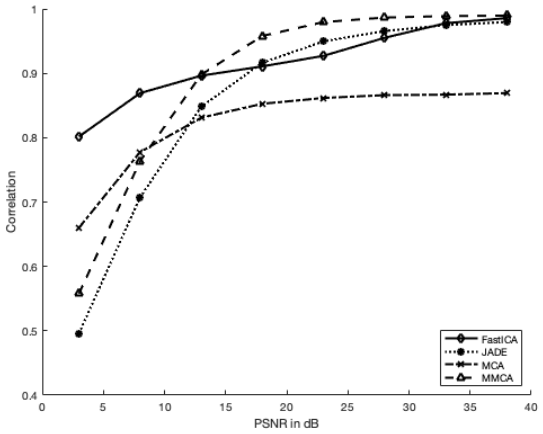


Figure 8: Evolution of the correlation coefficient between original and estimated sources as the noise variance varies.

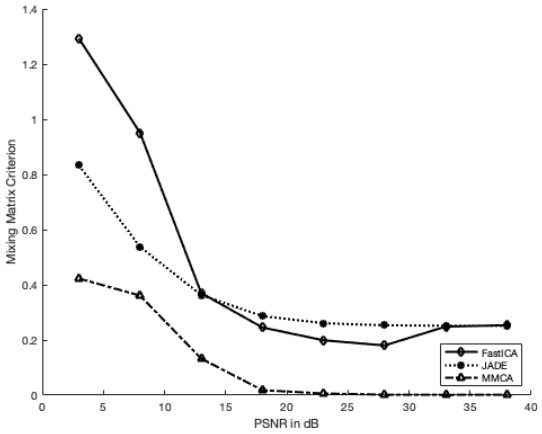


Figure 9: Evolution of the mixing matrix criterion (after indeterminacy corrected) as the noise variance varies.

4.4 Blind image source separation

In this experiment, we mix 4 pictures into 10 channels. The source images are picked as they contains similar morphologies. Classical ICA methods and MMCA, GMCA are applied to separate the source. Unfortunately the FastICA methods is not able to separate the original source. We hence adopt a sophisticated variant of it abbreviated as EFICA. Moreover, it has been proved in [5] that using a single overcomplete DWT dictionary or a union of DCT and DWT dictionary in GMCA provides similar results. We therefore use a discrete wavelet dictionary in FastGMCA (FGMCA).

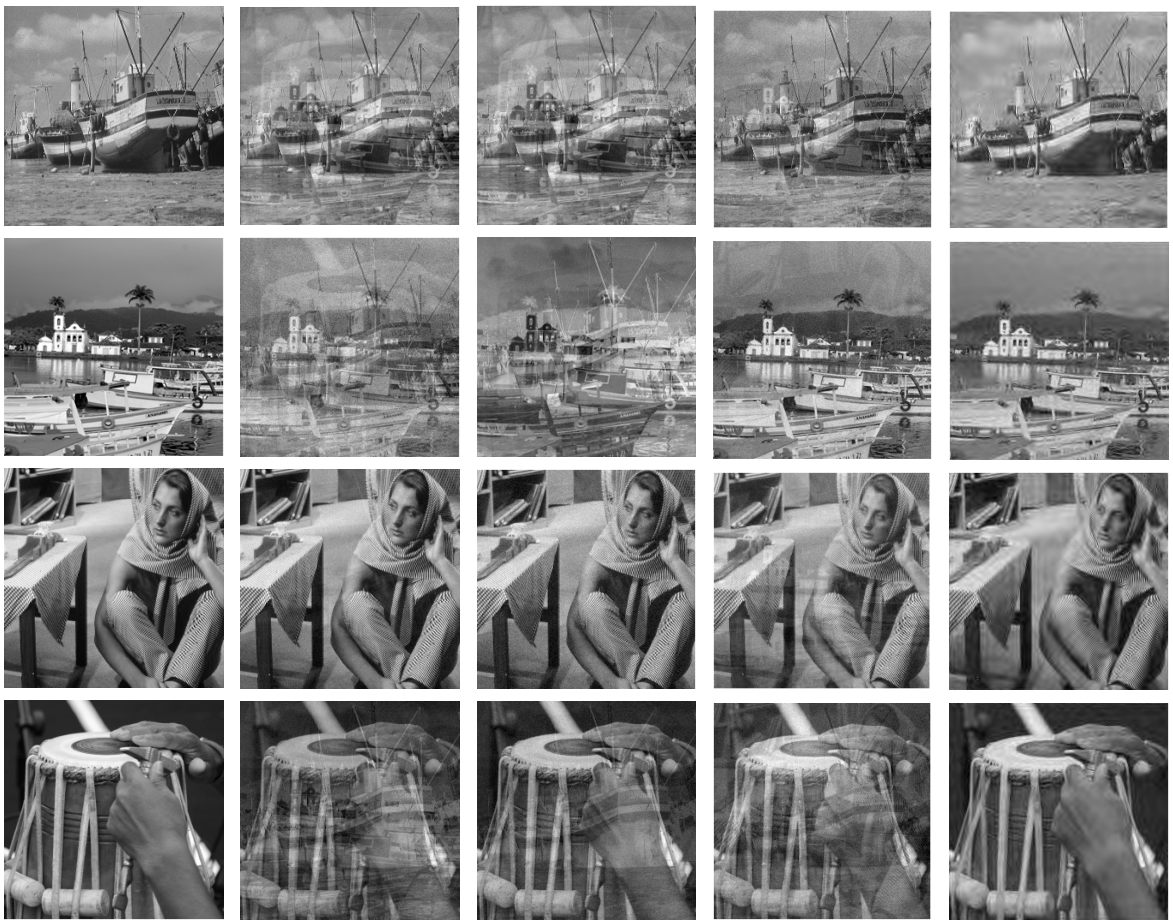
GMCA has been proved elsewhere [5] that it achieves better separation quality when the estimated components are 'very' sparse in the given dictionary. Therefore GMCA is supposed to perform well on separating images even without distinct discrepancies. Separated images are displayed in Figure (10). It shows the recovered image by various BSS methods contaminated by 20dB noise. All methods can distinguish the barbara image (row 3) from the mixtures. The ICA methods, nonetheless fail to separate the scenery photos (row 1 and 2). But in (e) using GMCA still gives acceptable result. MMCA using curvelet + DCT in (d) gives better result than ICA methods, but not so good as GMCA.

Figure (15) portrays the evolution of average correlation coefficient over 4 estimated sources as a function of the noise variance. At a first glance, GMCA significantly outperforms the ICA methods in terms of robustness and separation quality. Moreover, JADE performs the worst in among all ICA based algorithms. Figure (15) depicts the behavior of the mixing matrix criterion as the noise decreases. The mixing matrix criterion also clearly revels the strength of GMCA method.

To summarise the findings in this experiment, GMCA does take good advantage of overcompleteness and morphological diversity. We can aslo claim that sparsity brings better results. Taking the advantages further, in next experiment, we explore how locally learned dictionary helps the blind source separation.

4.5 Blind image separation using adaptive dictionary learning

In this experiment, simulations are provided to demonstrate the performance of the proposed BK-SVD algorithms, as compared with the baseline algorithms, K-SVD. Same as last experi-



(a) (b) (c) (d) (e)

Figure 10: Experiment2: image separation (20dB noise); **(a)**:Original image sources; **(b)**:EFICA outputs; **(c)**:JADE outputs; **(d)**:MMCA outputs; **(2)**:GMCA outputs.

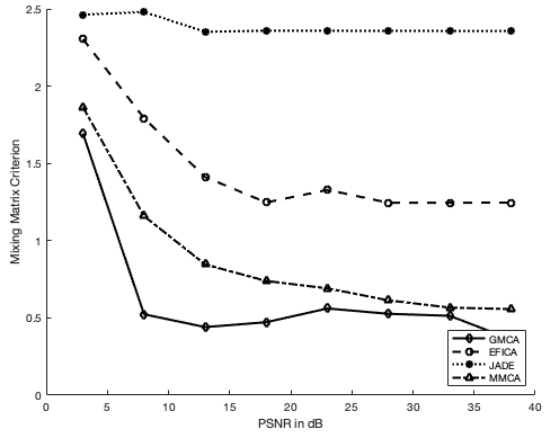


Figure 11: Evolution of the mixing matrix criterion (after indeterminacy corrected) as the noise variance varies.

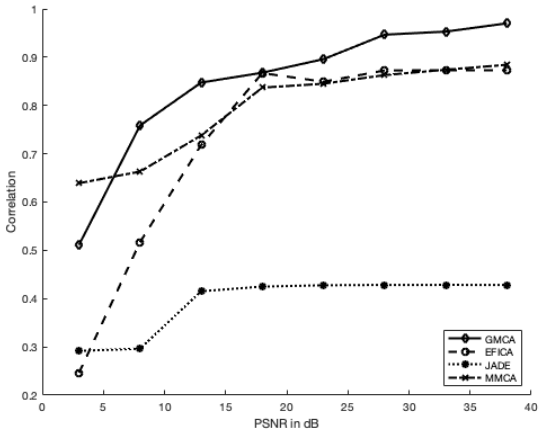
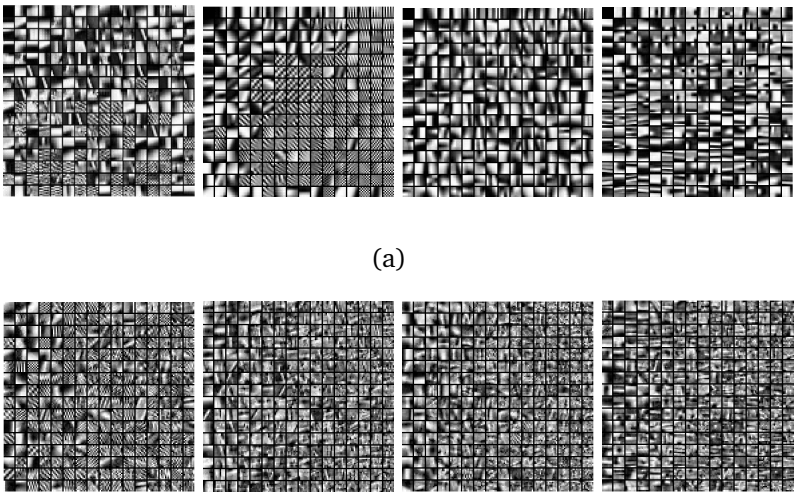


Figure 12: Evolution of the correlation coefficient between original and estimated sources as the noise variance varies.

ment settings, a severe case of 4 image sources with very different morphologies was chosen to exam the performance of the methods. 400 iterations were selected as the stopping criterion, additive Gaussian noise is added from 3dB to 40dB. The wavelet based GMCA is selected as the baseline algorithm.

In order to obtain enough training samples for dictionary learning, multiple overlapped segments (patches) of the sources are taken [1]. Choosing the optimal patch size is a subtle problem. Generally, very large patches should be avoided as they lead to massive dictionaries and also provide few training samples for the dictionary learning stage. We choose 8×8 patches for this experiment. Furthermore the patches are 50% overlapped as suggested in [1]. Maximal block size is set to be $s = 3$ and the block sparsity is $k = 2$ for BK-SVD+SAC. A standard DCT dictionary is chosen during initialisation. All dictionaries obtained have size of 64×256. Consequently, each dictionary in (13) have $16 \times 16 = 256$ blocks whereas each block is obtained from a 8×8 patches. Figure (13) also illustrates that both K-SVD and the proposed BK-SVD have good adaption to the corresponding sources, and looks significantly different from the standard DCT dictionary.



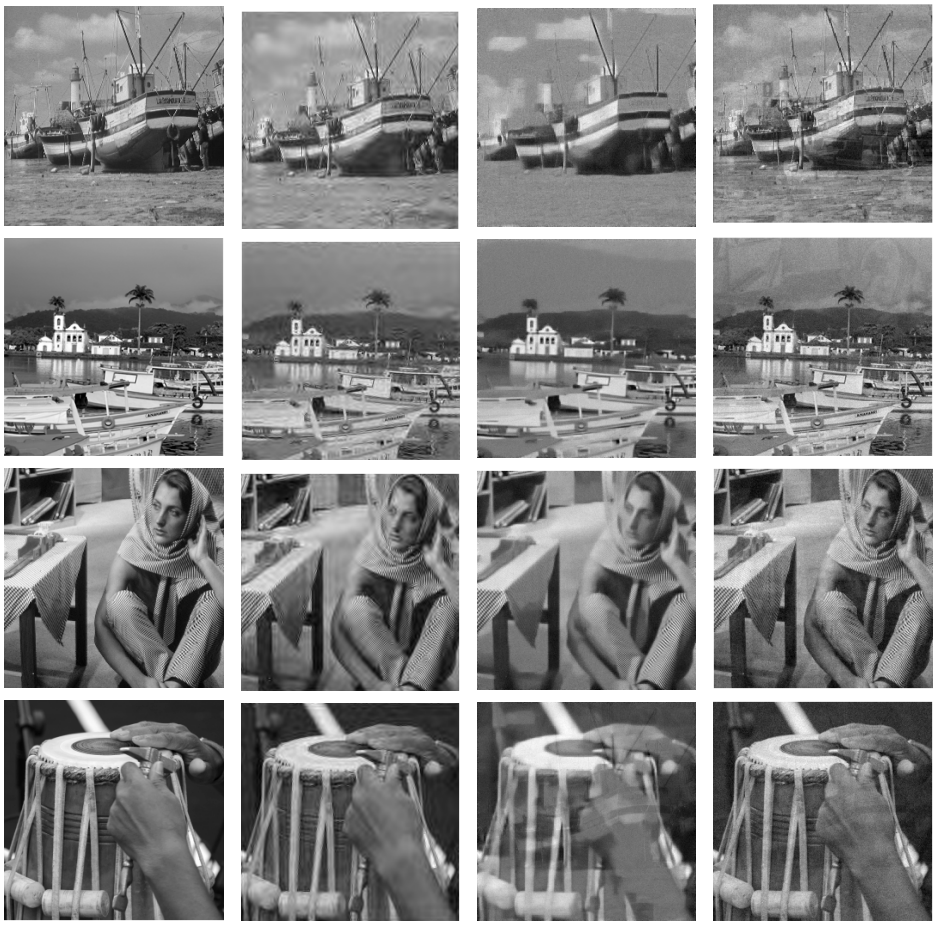
(a)

(b)

Figure 13: (a): Trained dictionary using K-SVD for sources 1,2,3,4 respectively; (b): Trained dictionary using SAC+BK-SVD for the 4 sources.

Figure (14) visually compares the separation results using several algorithm under 20dB Gaussian noise ($\sigma = 15$). Visually we can see that both two adaptive dictionary learning methods outperform the GMCA method in terms of recovered image quality. The learned dictionary helps to restore more details of the original image. Moreover, compared to the

proposed methods, separated image in K-SVD+MMCA is a bit blurry. We think 400 iterations maybe too large for this problem and the algorithm overfits. Afterall, the proposed method will show superiority to GMCA and K-SVD+MMCA in next paragraph.



(a) (b) (c) (d)

Figure 14: Experiment 3 (20dB noise): Using adaptive dictionary learning; **(a)**: original sources; **(b)**:GMCA outputs; **(c)**:K-SVD+MMCA output; **(d)**:BK-SVD+SAC+MMCA outputs;

We have shown via visual results that the block-structure dictionary learning algorithm provides convincing contribution to the blind source separation. Moreover, both the correlation and representation error proves our dictionary design method gives better performance. In Figure (15), for PSNR smaller than 13dB, the proposed block-sparsifying BSS method yields similar correlation coefficient as the K-SVD method and GMCA. This is because when the SNR is low, the algorithm may no longer successfully build a appropriate block structure. For low noise settings (PSNR is high), the proposed method clearly outperforms the other two methods. All three methods performs well in solving the mixing matrix in low noise settings, where BK-SVD behaves slightly better. Furthermore, Figure (17) plots the total reconstruction

error computed as $\|\mathbf{Y} - \mathbf{AX}\|_2$. We can see that the proposed SAC+BK-SVD leads to smaller errors than the K-SVD method.

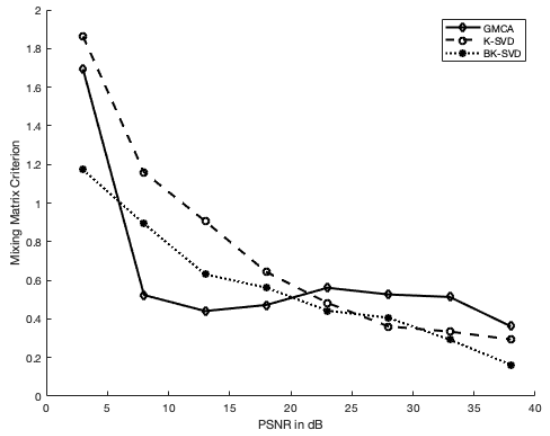


Figure 15: Evolution of the correlation coefficient between original and estimated sources as the noise variance varies.

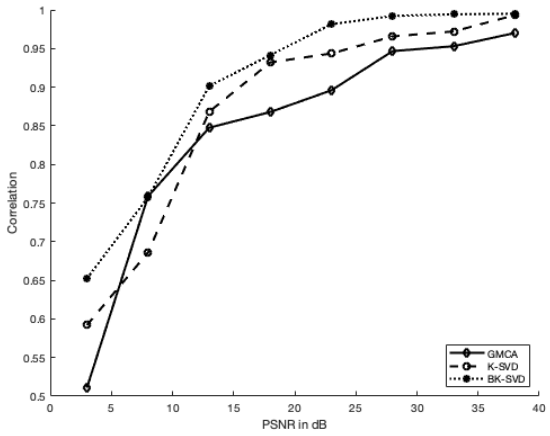


Figure 16: Evolution of the mixing matrix criterion (after indeterminacy corrected) as the noise variance varies.

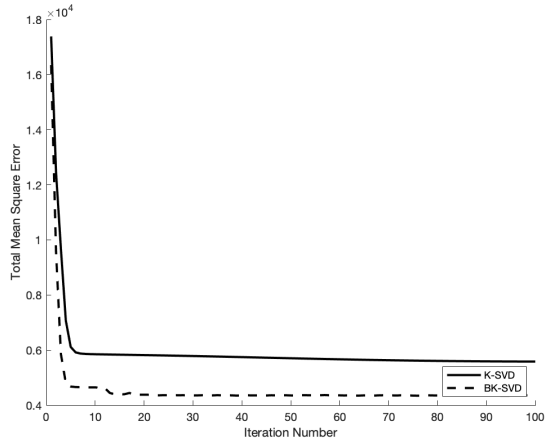


Figure 17: Total MSE versus number of iterations, Note that the elements of image sources have amplitude in the range $[0, 255]$.

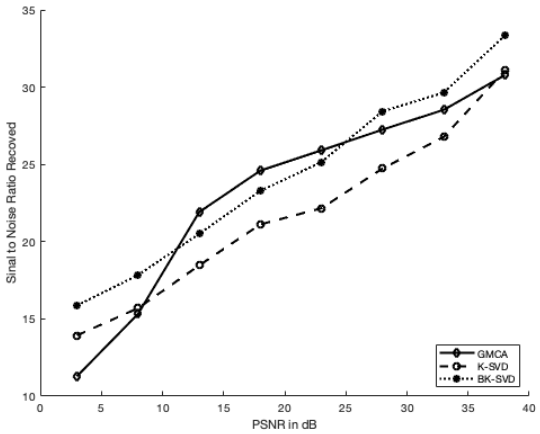


Figure 18: Average PSNR of the estimated sources as the Gaussian disturbance changes.

The running time for K-SVD BSS algorithm is 238 seconds. However the running time for BK-SVD BSS algorithm is 389 seconds. For Fast GMCA it is only 4 seconds. This convinces our deduction before, as the sparse agglomerative clustering algorithm dominates the complexity. Even though the BK-SVD algorithm is supposed to be faster than K-SVD, but the clustering stage slows down the overall process. It is also seen that, due to learning the dictionary, both algorithms are computationally demanding for large scale applications (i.e. image pro-

cessing) and are much slower than the GMCA algorithm. This implies that further effort is required to speed up the dictionary learning part in BSS.

MMCA	Fast GMCA	K-SVD	BK-SVD+SAC
$221s$	$4s$	$238s$	$389s$

Table 1: Running time for GMCA, KSVD and SAC+BK-SVD respectively, up to 100 iterations

In this section we examed the proposed BSS framework for the design of a block-sparsifying dictionary given a set of images and a maximal block size. Reuslts shown that it outperforms the prevailing K-SVD BSS algorithm by separation quality and reconstruction accuracy.

4.6 Choosing the best maximal block size and block sparsity level

From our experience in last section, we found that finding the optimal selection of best maximal block size s and block sparsity level (number of atoms in a block) k in SAC + B-KSVD is an unclear problem which needs more investigation. To avoid the long running time in BSS framework, we reduce the problem to a simple image reconstruction problem, where an image is given and the algorithm is intended to learn the dictionary and representation coefficients. Reconstruction quality is assessed by the mean square error (MSE). Similar as last experiment, we extract 8×8 patches from a 256×256 barbara image and set the maximum number of iterations to be 100. A 64×96 dictionary is learned and the sparse representation has dimension of 96×1024. Hence we can recover and resize the 64×1024 image. The MSE result of the proposed methods is again, compared with K-SVD.

We vary s and k from 1 to 6 and plot the reconstruction error using two methods in heatmaps. It is obvious that there exists a general trend that the reconstruction error decreases as the block size gets larger. It is reasonable because the more atoms are collected, the better will be the reconstruction quality. But the decreasing error trend is not monotonic for SAC+BK-SVD in Figure (20). For instance, the selection of $s = 4$ and $k = 5$ gives better outcome than $s = 5$ and $k = 5$. This means blindly increasing the parameters (more atoms numbers) is not an wise choice in SAC+BK-SVD. Furthermore, for $s \geq 6$ the superiority of the proposed methods no longer holds. To conclude, selecting block size and sparsity level needs discussion on a case by case basis, and is sometimes compromised with the runing time.

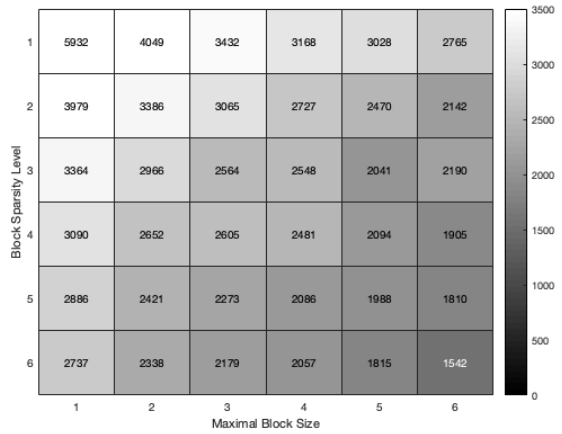


Figure 19: K-SVD reconstruction error heatmap against block size and sparsity level.

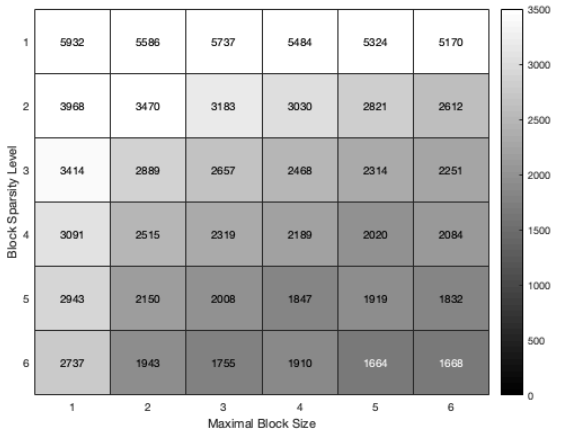


Figure 20: BK-SVD reconstruction error heatmap against block size and sparsity level.

5 Conclusion and futurework

In this report, different approaches for blind source separation are investigates and discussed. Then a sparse clustering based block dictionary learning algorithm is applied to the BSS problem. In every iteration of the BSS process, the proposed algorithm repeat two stages, a block structure clustering step (SAC) and a dictionary update setp (BK-SVD). When the maximal block size in SAC is reduced to 1, the proposed algorithm reverts to normal K-SVD. In contrasting to the normal K-SVD dictionary learning BSS algorithm, the proposed one is noted to give a sparser representation of the target image and exhibit a better estimation of the mixing matrix and sources.

However the proposed methods has certain limitations. The computation cost of our proposed method is not satisfactory. This is due to the cubed complexity of the SAC algorithm and blockwise updating manner of BK-SVD algorithm. In the future, we may consider the SimCo method [10] in computing the dictionary atoms. SimCo allows updating all codewords and all sparse coefficients simultaneously and is expected to significantly speed up the atom updating process. To further improve the proposed adaptive BSS methods, one could try and make the dictionary learning step less susceptible to local minimum traps. In addition, training one dictionary to sparsely represent all the sources is an alternative to calculating multiple distinct dictionaries, as long as the dictionary redundancy is large enough. An obvious advantage of using one dictionary is that the computational cost does not increase when the number of sources increases. Another refinement could be replacing blocks in the dictionary that contributes little to signal representations with the least significant signal elements.

This is expected to further improve the reconstruction ability of out dictionary learning algorithm. Besides, extending the proposed block-sparse BSS framework to underdetermined blind separation cases is also a valuable research direction.

References

[1] V. Abolghasemi, S. Ferdowsi, and S. Sanei. Blind separation of image sources via adaptive dictionary learning. *IEEE Transactions on Image Processing*, 21(6):2921–2930, June 2012.

[2] M Aharon, M Elad, and A Bruckstein. K-svd: An algorithm for designing overcomplete dictionaries for sparse representation. *IEEE Transactions on Signal Processing*, 54(11):4311–4322, 2006.

[3] Thomas Blumensath and Mike E. Davies. Iterative hard thresholding for compressed sensing. *CoRR*, abs/0805.0510, 2008.

[4] J. Bobin, Y. Moudden, J.-L. Starck, and M. Elad. Morphological diversity and source separation. *Signal Processing Letters, IEEE*, 13(7):409–412, 2006.

[5] J. Bobin, J.-L. Starck, J. Fadili, and Y. Moudden. Sparsity and morphological diversity in blind source separation. *Image Processing, IEEE Transactions on*, 16(11):2662–2674, 2007.

[6] Pau Bofill and Michael Zibulevsky. Underdetermined blind source separation using sparse representations. *Signal Processing*, 81(11):2353 – 2362, 2001.

[7] J. . Cardoso. Blind signal separation: statistical principles. *Proceedings of the IEEE*, 86(10):2009–2025, Oct 1998.

[8] Scott Shaobing Chen, David L. Donoho, and Michael A. Saunders. Atomic decomposition by basis pursuit. *SIAM Rev.*, 43(1):129–159, January 2001.

[9] W. Dai and O. Milenkovic. Subspace pursuit for compressive sensing signal reconstruction. *IEEE Transactions on Information Theory*, 55(5):2230–2249, May 2009.

[10] W. Dai, T. Xu, and W. Wang. Simultaneous codeword optimization (simco) for dictionary update and learning. *IEEE Transactions on Signal Processing*, 60(12):6340–6353, Dec 2012.

[11] Richard O. Duda, Peter E. Hart, and David G. Stork. *Pattern Classification (2Nd Edition)*. Wiley-Interscience, 2000.

[12] M. Elad, J.-L. Starck, P. Querre, and D.L. Donoho. Simultaneous cartoon and texture image inpainting using morphological component analysis (mca). *Applied and Computational Harmonic Analysis*, 19(3):340 – 358, 2005. Computational Harmonic Analysis - Part 1.

[13] P. Georgiev, F. Theis, and A. Cichocki. Sparse component analysis and blind source separation of underdetermined mixtures. *IEEE Transactions on Neural Networks*, 16(4):992–996, July 2005.

[14] A. Hyvarinen. Fast ica for noisy data using gaussian moments. In *ISCAS'99. Proceedings of the 1999 IEEE International Symposium on Circuits and Systems VLSI (Cat. No.99CH36349)*, volume 5, pages 57–61 vol.5, May 1999.

[15] A. Hyvrinen and E. Oja. Independent component analysis: algorithms and applications. *Neural Networks*, 13(4):411 – 430, 2000.

[16] Stephen C. Johnson. Hierarchical clustering schemes. *Psychometrika*, 32(3):241–254, Sep 1967.

[17] S. G. Mallat and Zhifeng Zhang. Matching pursuits with time-frequency dictionaries. *IEEE Transactions on Signal Processing*, 41(12):3397–3415, Dec 1993.

[18] E. Oja, K. Kiviluoto, and S. Malaroiu. Independent component analysis for financial time series. *Adaptive Systems for Signal Processing, Communications, and Control Symposium 2000. AS-SPCC. The IEEE 2000*, pages 111–116, 2000.

[19] K. Raju, T. Ristaniemi, J. Karhunen, and E. Oja. Jammer suppression in ds-cdma arrays using independent component analysis. *IEEE Transactions on Wireless Communications*, 5(1):77–82, Jan 2006.

[20] Ron Rubinstein, Michael Zibulevsky, and Michael Elad. Efficient implementation of the k-svd algorithm using batch orthogonal matching pursuit. *CS Technion*, 40, 01 2008.

[21] Sylvain Sardy, Andrew G.Bruce, and Paul Tseng. Block coordinate relaxation methods for nonparametric wavelet denoising. *Journal of Computational and Graphical Statistics*, 9(2):361–379, 2000.

[22] J. L. Starck, Yassir Moudden, Jérôme Bobin, Michael Elad, and David L. Donoho. Morphological component analysis. 2005.

[23] P. Tichavsky and Z. Koldovsky. Optimal pairing of signal components separated by blind techniques. *IEEE Signal Processing Letters*, 11(2):119–122, Feb 2004.

[24] H. Xu, N. Fu, C. Yin, L. Qiao, and X. Peng. Blind separation of sufficiently sparse sources in multichannel compressed sensing. pages 515–520, Aug 2014.

[25] L. Zelnik-Manor, K. Rosenblum, and Y. C. Eldar. Dictionary optimization for block-sparse representations. *IEEE Transactions on Signal Processing*, 60(5):2386–2395, May 2012.

[26] Michael Zibulevsky and Barak A. Pearlmutter. Blind source separation by sparse decomposition in a signal dictionary. *Neural Computation*, 13(4):863–882, 2001.