

# AN ICA ALGORITHM FOR ANALYZING MULTIPLE DATA SETS

Ana S. Lukic<sup>1</sup>, Miles N. Wernick<sup>1</sup>, Lars Kai Hansen<sup>2</sup>, Stephen C. Strother<sup>3</sup>

<sup>1</sup>Illinois Institute of Technology, Chicago, <sup>2</sup>Technical University of Denmark, Lyngby, Denmark

<sup>3</sup>VA Medical Center and University of Minnesota, Minneapolis

## ABSTRACT

*In this paper we derive an independent-component analysis (ICA) method for analyzing two or more data sets simultaneously. Our model permits there to be components individual to the various data sets, and others that are common to all the sets. We explore the assumed time autocorrelation of independent signal components and base our algorithm on prediction analysis. We illustrate the algorithm using a simple image separation example. Our aim is to apply this method to functional brain mapping using functional magnetic resonance imaging (fMRI).*

## 1. INTRODUCTION

There are several approaches to the problem of blind separation of linear signal mixtures. Some of them are based on non-Gaussianity [1, 3] while others exploit temporal correlations [2, 5]. The technique of Molgedey and Schuster [5] is especially attractive since it offers a non-iterative solution. The limitations of the original algorithm, such as restriction to square non-singular mixing matrices and inherent erroneous complex-valued results, were eliminated in [4].

In this paper, we extend the Molgedey-Schuster ICA method to analyze more than one data set simultaneously. There are many potential applications for simultaneous independent-component analysis (ICA) of multiple data sets. For example, one might simultaneously analyze the audio and video portions of a video sequence.

Our interest is in developing methods to analyze functional magnetic resonance images (fMRI) of the brain, to map spatial and temporal patterns of brain activation. In this application, the data can be divided into multiple data sets by considering different subjects separately, or by considering image data separately from other measurements made during the fMRI study (e.g., reaction time of the subject).

## 2. EXTENDED BLIND SIGNAL SEPARATION PROBLEM

In the classical blind signal separation problem it is assumed that there are  $N$  independent source signals in matrix  $\mathbf{S}$  observed through a mixing matrix  $\mathbf{A}$ , i.e.,

$$\mathbf{X} = \mathbf{A}\mathbf{S}, \quad (1)$$

to obtain a matrix of observations  $\mathbf{X}$ . The goal in this classical ICA problem is to recover the sources in  $\mathbf{S}$  and the mixing matrix  $\mathbf{A}$  without any prior knowledge except the assumption of independence of the sources in  $\mathbf{S}$ .

To analyze two data sets  $\mathbf{X}$  and  $\mathbf{Y}$  simultaneously, we could lump them together into one matrix, and use the model in (1); however, this would force the description of the two matrix to be based on the same set of sources. To allow the two data sets to have some common components, and other distinct components, we propose the following model:

$$\begin{aligned} \mathbf{X} &= \mathbf{A}_x \mathbf{S}_x + \mathbf{B}_x \mathbf{T} \\ \mathbf{Y} &= \mathbf{A}_y \mathbf{S}_y + \mathbf{B}_y \mathbf{T} \end{aligned} \quad (2)$$

where  $\mathbf{X}$  and  $\mathbf{Y}$  are two sets of observations;  $\mathbf{S}_x$  and  $\mathbf{S}_y$  are independent sources specific to  $\mathbf{X}$  and  $\mathbf{Y}$ , respectively;  $\mathbf{T}$  is a matrix of  $K$  independent sources common to both  $\mathbf{X}$  and  $\mathbf{Y}$ ;  $\mathbf{A}_x$ ,  $\mathbf{A}_y$ ,  $\mathbf{B}_x$ , and  $\mathbf{B}_y$  are mixing matrices. The numbers of observations in  $\mathbf{X}$  and  $\mathbf{Y}$  are  $L$  and  $P$ , respectively; the numbers of sources in  $\mathbf{S}_x$  and  $\mathbf{S}_y$  are  $M$  and  $N$ , respectively. In this paper, we discuss the algorithm in terms of only two data sets, but it is easily extended to more than two.

## 3. MIXING MATRICES AS EIGENVECTOR MATRICES

In this section we show that the mixing matrices in this ICA problem can be obtained as eigenvectors of linear minimum mean-square error (LMMSE) prediction matrices.

Assuming the independent signals have non-vanishing time autocorrelation functions we suppose there is a LMMSE prediction matrix  $\mathbf{W}$  such that:

$$\begin{bmatrix} \mathbf{X}_\tau \\ \mathbf{Y}_\tau \end{bmatrix} = \mathbf{W} \begin{bmatrix} \mathbf{X} \\ \mathbf{Y} \end{bmatrix} \quad (3)$$

where  $\mathbf{X}_\tau$  and  $\mathbf{Y}_\tau$  are delayed versions of  $\mathbf{X}$  and  $\mathbf{Y}$  respectively. By combining (2) and (3) we can write:

$$\begin{bmatrix} \mathbf{A}_x \mathbf{S}_{x\tau} + \mathbf{B}_x \mathbf{T}_\tau \\ \mathbf{A}_y \mathbf{S}_{y\tau} + \mathbf{B}_y \mathbf{T}_\tau \end{bmatrix} = \begin{bmatrix} \mathbf{W}_{xx} & \mathbf{W}_{xy} \\ \mathbf{W}_{yx} & \mathbf{W}_{yy} \end{bmatrix} \begin{bmatrix} \mathbf{A}_x \mathbf{S}_x + \mathbf{B}_x \mathbf{T} \\ \mathbf{A}_y \mathbf{S}_y + \mathbf{B}_y \mathbf{T} \end{bmatrix} \quad (4)$$

where we have divided the matrix  $\mathbf{W}$  into four blocks  $\mathbf{W}_{xx}$ ,  $\mathbf{W}_{xy}$ ,  $\mathbf{W}_{yx}$  and  $\mathbf{W}_{yy}$ ;  $\mathbf{S}_{x\tau}$ ,  $\mathbf{S}_{y\tau}$  and  $\mathbf{T}_\tau$  are delayed versions of  $\mathbf{S}_x$ ,  $\mathbf{S}_y$  and  $\mathbf{T}$ , respectively.

Right multiplying (4) by  $\mathbf{S}_x^T$  yields:

$$\begin{bmatrix} \mathbf{A}_x \mathbf{S}_{x\tau} \mathbf{S}_x^T + \mathbf{B}_x \mathbf{T}_\tau \mathbf{S}_x^T \\ \mathbf{A}_y \mathbf{S}_{y\tau} \mathbf{S}_x^T + \mathbf{B}_y \mathbf{T}_\tau \mathbf{S}_x^T \end{bmatrix} = \begin{bmatrix} \mathbf{W}_{xx} & \mathbf{W}_{xy} \\ \mathbf{W}_{yx} & \mathbf{W}_{yy} \end{bmatrix} \begin{bmatrix} \mathbf{A}_x \mathbf{S}_x \mathbf{S}_x^T + \mathbf{B}_x \mathbf{T} \mathbf{S}_x^T \\ \mathbf{A}_y \mathbf{S}_y \mathbf{S}_x^T + \mathbf{B}_y \mathbf{T} \mathbf{S}_x^T \end{bmatrix} \quad (5)$$

Because the sources in  $\mathbf{S}_x$ ,  $\mathbf{S}_y$  and  $\mathbf{T}$  are independent,  $\mathbf{S}_{y\tau} \mathbf{S}_x^T = 0$ ,  $\mathbf{S}_y \mathbf{S}_x^T = 0$ ,  $\mathbf{T}_\tau \mathbf{S}_x^T = 0$  and  $\mathbf{T} \mathbf{S}_x^T = 0$ . Thus, (5) becomes:

$$\begin{bmatrix} \mathbf{A}_x \mathbf{C}_x(\tau) \\ 0 \end{bmatrix} = \begin{bmatrix} \mathbf{W}_{xx} & \mathbf{W}_{xy} \\ \mathbf{W}_{yx} & \mathbf{W}_{yy} \end{bmatrix} \begin{bmatrix} \mathbf{A}_x \mathbf{C}_x(0) \\ 0 \end{bmatrix}, \quad (6)$$

where  $\mathbf{C}_x(\tau) = \mathbf{S}_{x\tau} \mathbf{S}_x^T$  and  $\mathbf{C}_x(0) = \mathbf{S}_x \mathbf{S}_x^T$ . The first row of (6) yields

$$\mathbf{W}_{xx} \mathbf{A}_x = \mathbf{A}_x [\mathbf{C}_x(\tau) \mathbf{C}_x(0)^{-1}] \quad (7)$$

which is an eigenvector equation for  $\mathbf{W}_{xx}$  since  $\mathbf{C}_x(\tau) \mathbf{C}_x(0)^{-1}$  is diagonal. The second row of (6) yields

$$0 = \mathbf{W}_{yx} \mathbf{A}_x \quad (8)$$

since  $\mathbf{C}_x(0)$  is a full rank positive diagonal matrix.

Similarly, by right-multiplying (4) by  $\mathbf{S}_y^T$  we obtain:

$$0 = \mathbf{W}_{xy} \mathbf{A}_y \mathbf{C}_y(0) \quad (9)$$

$$\mathbf{W}_{yy} \mathbf{A}_y = \mathbf{A}_y [\mathbf{C}_y(\tau) \mathbf{C}_y(0)^{-1}] \quad (10)$$

Finally, after right multiplying (4) by  $\mathbf{T}^T$  we obtain:

$$\begin{bmatrix} \mathbf{B}_x \\ \mathbf{B}_y \end{bmatrix} \mathbf{C}_T(\tau) = \begin{bmatrix} \mathbf{W}_{xx} & \mathbf{W}_{xy} \\ \mathbf{W}_{yx} & \mathbf{W}_{yy} \end{bmatrix} \begin{bmatrix} \mathbf{B}_x \\ \mathbf{B}_y \end{bmatrix} \mathbf{C}_T(0) \quad (11)$$

or, rearranging,

$$\mathbf{W} \begin{bmatrix} \mathbf{B}_x \\ \mathbf{B}_y \end{bmatrix} = \begin{bmatrix} \mathbf{B}_x \\ \mathbf{B}_y \end{bmatrix} [\mathbf{C}_T(\tau) \mathbf{C}_T(0)^{-1}] \quad (12)$$

Equations (7), (10), and (12) are eigenvector equations that provide a recipe for identifying the various mixing matrices as eigenvectors of prediction matrices.

#### 4. THE ALGORITHM

We divide the algorithm in two parts. First, we estimate the prediction matrix  $\mathbf{W}$  and then we use it to estimate mixing matrices and independent sources. It is well known [6] that the LMMSE prediction matrix is:

$$\hat{\mathbf{W}}(\tau) = \hat{\mathbf{R}}_{\mathbf{XY}}(\tau) \hat{\mathbf{R}}_{\mathbf{XY}}(0)^{-1}, \quad (13)$$

where  $\hat{\mathbf{R}}_{\mathbf{XY}}(0)$  is an estimate of the autocorrelation function at lag 0,

$$\hat{\mathbf{R}}_{\mathbf{XY}}(0) = \begin{bmatrix} \mathbf{X} \\ \mathbf{Y} \end{bmatrix} \begin{bmatrix} \mathbf{X}^T & \mathbf{Y}^T \end{bmatrix}, \quad (14)$$

and  $\hat{\mathbf{R}}_{\mathbf{XY}}(\tau)$  is an estimate of the autocorrelation function at lag  $\tau$ ,

$$\hat{\mathbf{R}}_{\mathbf{XY}}(\tau) = \frac{1}{2} \left( \begin{bmatrix} \mathbf{X}_\tau \\ \mathbf{Y}_\tau \end{bmatrix} \begin{bmatrix} \mathbf{X}^T & \mathbf{Y}^T \end{bmatrix} + \begin{bmatrix} \mathbf{X} \\ \mathbf{Y} \end{bmatrix} \begin{bmatrix} \mathbf{X}_\tau^T & \mathbf{Y}_\tau^T \end{bmatrix} \right) \quad (15)$$

where  $\mathbf{X}_\tau$  and  $\mathbf{Y}_\tau$  are cyclic permutations by  $\tau$  steps, *i.e.*  $\mathbf{X}_\tau = \{\mathbf{x}((n+\tau)_T)\}$ ,  $\mathbf{x}(k)$  denotes the  $k^{\text{th}}$  column of  $\mathbf{X}$  and  $(\cdot)_T$  denotes the argument modulo  $T$ . Equation (15) guarantees that  $\hat{\mathbf{R}}_{\mathbf{XY}}(\tau)$  is symmetric. Consider the singular value decompositions (SVDs):

$$\begin{bmatrix} \mathbf{X} \\ \mathbf{Y} \end{bmatrix} = \mathbf{U} \mathbf{D} \mathbf{V}^T \quad (16)$$

$$\begin{bmatrix} \mathbf{X}_\tau \\ \mathbf{Y}_\tau \end{bmatrix} = \mathbf{U}_\tau \mathbf{D}_\tau \mathbf{V}_\tau^T \quad (17)$$

where eigenvector matrices contain only components whose corresponding eigenvalues are nonzero. Since  $\mathbf{X} \mathbf{X}^T = \mathbf{X}_\tau \mathbf{X}_\tau^T$  it follows that  $\mathbf{U}_\tau = \mathbf{U}$  and  $\mathbf{D}_\tau = \mathbf{D}$ . Then:

$$\begin{aligned} \hat{\mathbf{W}}(\tau) &= \frac{1}{2} \left( \mathbf{U} \mathbf{D} \mathbf{V}_\tau^T (\mathbf{U} \mathbf{D} \mathbf{V}^T)^T + \mathbf{U} \mathbf{D} \mathbf{V}^T (\mathbf{U} \mathbf{D} \mathbf{V}_\tau^T)^T \right) (\mathbf{U} \mathbf{D} \mathbf{V}^T (\mathbf{U} \mathbf{D} \mathbf{V}^T)^T)^{-1} \\ &= \frac{1}{2} \mathbf{U} \mathbf{D} (\mathbf{V}_\tau^T \mathbf{V} + \mathbf{V}^T \mathbf{V}_\tau) \mathbf{D}^{-1} \mathbf{U}^T \end{aligned} \quad (18)$$

We will estimate  $\hat{\mathbf{W}}$  as the average of  $\hat{\mathbf{W}}(\tau)$  over a set of values for  $\tau$ :

$$\hat{\mathbf{W}} = \frac{1}{t} \sum_{j=1}^t \hat{\mathbf{W}}(\tau_j) = \frac{1}{2t} \mathbf{U} \mathbf{D} \left( \sum_{j=1}^t (\mathbf{V}_{\tau_j}^T \mathbf{V} + \mathbf{V}^T \mathbf{V}_{\tau_j}) \right) \mathbf{D}^{-1} \mathbf{U}^T \quad (19)$$

where  $t$  is the number of different values of  $\tau$  we want to average over and  $\tau_j, j=1 \dots t$  are those values. We are now ready to outline the algorithm:

PART I: Calculating prediction matrix  $\hat{\mathbf{W}}$

*Step 1:* Perform the SVD in (16), keep only the components whose eigenvalues are nonzero.

*Step 2:* Calculate  $\mathbf{K} = \sum_{j=1}^t (\mathbf{V}_{\tau_j}^T \mathbf{V} + \mathbf{V}^T \mathbf{V}_{\tau_j})$

*Step 3:* Calculate  $\hat{\mathbf{W}} = \frac{1}{2t} \mathbf{U} \mathbf{D} \mathbf{K} \mathbf{D}^{-1} \mathbf{U}^T$

PART II: Estimating mixing matrices

*Step 1:* Estimate  $\hat{\mathbf{A}}_x$  by selecting only the eigenvectors of  $\mathbf{W}_{xx}$  satisfying (8) and having corresponding nonzero eigenvalues. Denote the number of such eigenvectors (columns of  $\hat{\mathbf{A}}_x$ ) by  $N$ .

*Step 2:* Estimate  $\hat{\mathbf{A}}_y$  by selecting only the eigenvectors of  $\mathbf{W}_{yy}$  satisfying (9) and having corresponding nonzero eigenvalues. Denote the number of such eigenvectors (columns of  $\hat{\mathbf{A}}_y$ ) by  $M$ .

*Step 3:* Estimate  $\hat{\mathbf{B}} = [\hat{\mathbf{B}}_x^T \hat{\mathbf{B}}_y^T]^T$  as the matrix of  $K$  eigenvectors of  $\mathbf{W}$  corresponding to the  $K$  largest

eigenvalues of  $\mathbf{W}$  where:

$$K = \text{rank}(\begin{bmatrix} \mathbf{X}^T & \mathbf{Y}^T \end{bmatrix}) - N - M \quad (20)$$

PART III: Estimating independent sources

Step 1: From  $\mathbf{X} = \begin{bmatrix} \mathbf{A}_x & \mathbf{B}_x \end{bmatrix} \begin{bmatrix} \mathbf{S}_x \\ \mathbf{T} \end{bmatrix}$  calculate:

$$\begin{bmatrix} \hat{\mathbf{S}}_x \\ \hat{\mathbf{T}}_x \end{bmatrix} = \left( \begin{bmatrix} \hat{\mathbf{A}}_x^T \\ \hat{\mathbf{B}}_x^T \end{bmatrix} \begin{bmatrix} \hat{\mathbf{A}}_x & \hat{\mathbf{B}}_x \end{bmatrix} \right)^{-1} \begin{bmatrix} \hat{\mathbf{A}}_x^T \\ \hat{\mathbf{B}}_x^T \end{bmatrix} \mathbf{X} \quad (21)$$

Step 2: Calculate:

$$\begin{bmatrix} \hat{\mathbf{S}}_y \\ \hat{\mathbf{T}}_y \end{bmatrix} = \left( \begin{bmatrix} \hat{\mathbf{A}}_y^T \\ \hat{\mathbf{B}}_y^T \end{bmatrix} \begin{bmatrix} \hat{\mathbf{A}}_y & \hat{\mathbf{B}}_y \end{bmatrix} \right)^{-1} \begin{bmatrix} \hat{\mathbf{A}}_y^T \\ \hat{\mathbf{B}}_y^T \end{bmatrix} \mathbf{Y} \quad (22)$$

Step 3: Calculate  $\hat{\mathbf{T}} = \frac{\hat{\mathbf{T}}_x + \hat{\mathbf{T}}_y}{2}$ , since  $\hat{\mathbf{T}}_x$  and  $\hat{\mathbf{T}}_y$  should be the same.

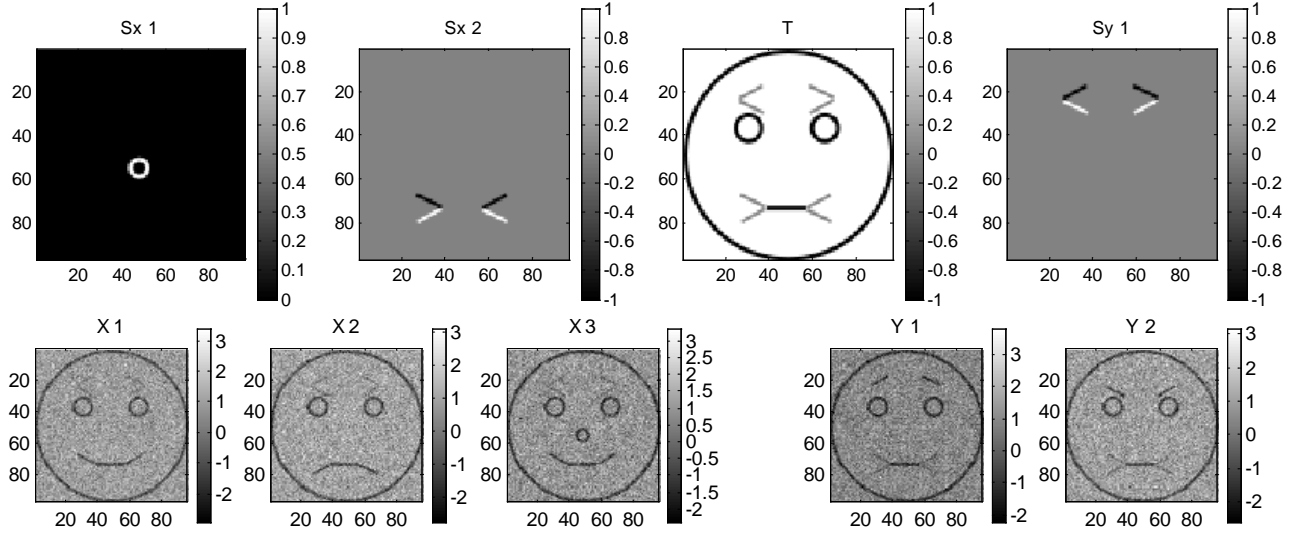
## 5. RESULTS

At this writing, we have nearly completed a demonstration of the method using fMRI data; however, the results are too preliminary to report and the submission deadline is upon us. We plan to have the analysis complete by the time of the conference.

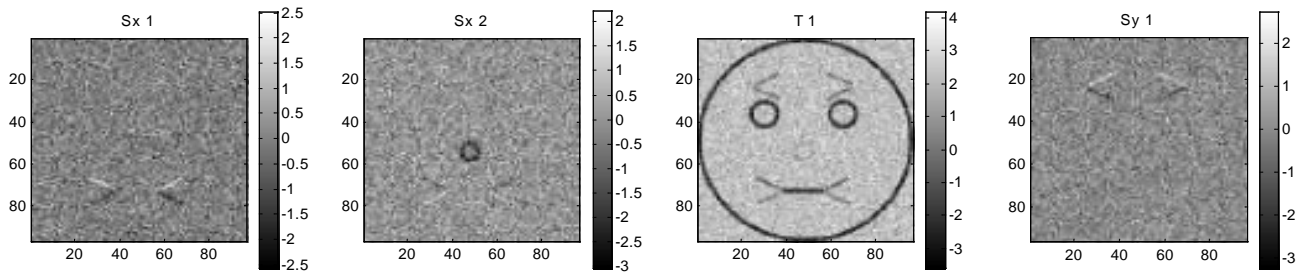
In the meantime, to illustrate the algorithm simply we extend the toy example given in [4]. The top row of Fig. 1 shows the four image components used to produce the data set in the bottom row:

$$\mathbf{X} = \begin{bmatrix} \mathbf{x}_1 \\ \mathbf{x}_2 \\ \mathbf{x}_3 \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 0 & -1 \\ -2 & 1 \end{bmatrix} \begin{bmatrix} \mathbf{s}_{x1} \\ \mathbf{s}_{x2} \end{bmatrix} + \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} \mathbf{t}_1 + \mathbf{N} \quad (23)$$

$$\mathbf{Y} = \begin{bmatrix} \mathbf{y}_1 \\ \mathbf{y}_2 \end{bmatrix} = \begin{bmatrix} 1 \\ -1 \end{bmatrix} \mathbf{s}_{y1} + \begin{bmatrix} 1 \\ 1 \end{bmatrix} \mathbf{t}_1 + \mathbf{N} \quad (24)$$



**Figure 1.** Top row shows four independent image components used to produce the images in the bottom row. Bottom row shows two sets of mixtures  $\mathbf{X}$  and  $\mathbf{Y}$  of the four components with white Gaussian noise added. The first two components appear only in the mixture  $\mathbf{X}$ , the third one is common to both while the last one is specific to  $\mathbf{Y}$  only



**Figure 2.** Results of the proposed algorithm applied to the  $\mathbf{X}$  and  $\mathbf{Y}$  mixtures shown in the bottom row of Fig. 1. In this reconstructed set of independent components the first two changed places. Also, the first, second and fourth components are inverted compared to the original set.

where  $\mathbf{x}_i, i=1..3$  and  $\mathbf{y}_i, i=1..2$  are two observed sets,  $\mathbf{s}_{x1}, \mathbf{s}_{x2}, \mathbf{t}_1, \mathbf{s}_{y1}$  are four independent image components and  $\mathbf{N}$  is the matrix of zero-mean Gaussian white noise with variance  $\sigma_n^2 = 0.3$ . The results of our algorithm are shown in Fig. 2. The estimated mixing matrices are:

$$\hat{\mathbf{A}}_x = \begin{bmatrix} -0.7260 & -0.3134 \\ 0.6862 & -0.4148 \\ 0.0443 & 0.8542 \end{bmatrix}, \quad \hat{\mathbf{A}}_y = \begin{bmatrix} -0.7142 \\ 0.6999 \end{bmatrix},$$

$$\hat{\mathbf{B}}_x = \begin{bmatrix} 0.4539 \\ 0.4475 \\ 0.4398 \end{bmatrix}, \quad \hat{\mathbf{B}}_y = \begin{bmatrix} 0.4482 \\ 0.4466 \end{bmatrix} \quad (25)$$

Comparing the original independent sources from Fig. 2 and those produced by the algorithm it can be seen that  $\hat{\mathbf{s}}_{x1}$  is the inverted and scaled version of  $\mathbf{s}_{x2}$ ,  $\hat{\mathbf{s}}_{x2}$  is the inverted and scaled version of  $\mathbf{s}_{x1}$ , and that  $\hat{\mathbf{s}}_{y1}$  is the inverted version of  $\mathbf{s}_{y1}$ . We can write:

$$\mathbf{X} = \begin{bmatrix} 0.6268 & 1.4520 \\ 0.8296 & -1.2724 \\ -1.6084 & -0.0886 \end{bmatrix} \begin{bmatrix} -\frac{1}{2}\hat{\mathbf{s}}_{x2} \\ \frac{1}{2}\hat{\mathbf{s}}_{x1} \end{bmatrix} + \begin{bmatrix} 0.9078 \\ 0.8950 \\ 0.8796 \end{bmatrix} \begin{bmatrix} \frac{1}{2}\hat{\mathbf{t}}_1 \\ \frac{1}{2}\hat{\mathbf{t}}_1 \end{bmatrix} \quad (26)$$

$$\mathbf{Y} = \begin{bmatrix} 1.0100 \\ -0.9898 \end{bmatrix} \begin{bmatrix} -\frac{1}{\sqrt{2}}\hat{\mathbf{s}}_{y1} \end{bmatrix} + \begin{bmatrix} 0.8964 \\ 0.8932 \end{bmatrix} \begin{bmatrix} \frac{1}{2}\hat{\mathbf{t}}_1 \\ \frac{1}{2}\hat{\mathbf{t}}_1 \end{bmatrix} \quad (27)$$

where the estimated mixing matrices are closer to the originals. This illustrates that the algorithm recovers the independent sources up to a scale and permutation, as expected.

## 6. DISCUSSION

The algorithm presented while simple in theory might be difficult to implement especially in the presence of noise. The condition crucial to determining the number of independent components in  $\mathbf{S}_x$  is (8). Due to the presence of noise and an error made by the prediction none of the columns in the product will be exactly zero. To overcome the problem we set the following threshold and took only the columns that satisfy:

$$\frac{\sum_{p=1}^P \left| \sum_{l=1}^L [\mathbf{W}_{yx}]_{p,l} [\mathbf{E}_x]_{l,i} \right|}{\sum_{i=1}^L \sum_{p=1}^P \left| \sum_{l=1}^L [\mathbf{W}_{yx}]_{p,l} [\mathbf{E}_x]_{l,i} \right|} < 0.04 \quad i = 1 \dots L \quad (28)$$

where  $\mathbf{E}_x$  is the eigen vector matrix of  $\mathbf{W}_{xx}$ . This means that we took the absolute value of  $\mathbf{W}_{yx} \mathbf{E}_x$ , calculated the sum of elements in each column, normalized by dividing by the overall sum of elements, and thresholded the resulting score. We repeated the same procedure to estimate  $\mathbf{A}_y$ .

Another important fact is that (20) does not hold in the presence of white noise so principal component analysis (PCA) is required as a preprocessing step, a method proven successful in [1]. In this example we first smoothed the data by performing the PCA on pooled data set  $[\mathbf{X}^T \ \mathbf{Y}^T]$  and eliminating the principal components whose eigenvectors do not satisfy:

$$h_i > \left(\frac{2.1}{n^2}\right)^2 \quad i = 1 \dots L+P \quad (29)$$

where  $h_i$  is the  $i^{\text{th}}$  eigen value and  $n^2$  is number of pixels in each image. To demonstrate the robustness of the algorithm to the noise we repeated the experiment with higher noise level,  $\sigma_n^2 = 1$ , and obtained the estimates of mixing matrices similar to those in (25).

## 7. CONCLUSIONS

In this paper we extended the blind source separation problem and presented an ICA algorithm based on non-vanishing source autocorrelation functions for solving it. We assumed that there are more than one set of observations and that independent sources can be common to all observations or specific to some observations only. The algorithm developed is an extension of the original Molgedey-Schuster algorithm and is capable of distinguishing between the independent components specific to particular observations and those common to all of them. It is very robust to the amount of white noise added although PCA is required as a preprocessing step. It can easily be extended to the case when there are more than two sets of observations, each containing a mixture of components common to all observations and components specific to the particular sets only.

## 8. REFERENCES

- [1] A. Bell and T. J. Sejnowski, "An Information-Maximization Approach to Blind Separation and Blind Deconvolution," *Neural Computation*, vol. 7, pp. 1129-1159, 1995.
- [2] Cardoso, J.-F., A. Belouchrani, K. Abed-Meriam, and E. Moulines, "Blind Source Separation Technique Using Second-Order Statistics," *IEEE Transactions on Signal Processing*, vol. 45, pp. 434-444, 1997.
- [3] Comon, P., "Independent Component Analysis: A New Concept?," *Signal Processing*, vol. 36, pp. 287-314, 1994.
- [4] Larsen, J., L. K. Hansen, and T. Kolenda, "On Independent Component Analysis for Multimedia Signals," in *Multimedia Image and Video Processing*, G. L., K. S.Y., and J. Larsen, Eds.: CRC Press, 2000, pp. 175-200.
- [5] Molgedey, L. and H. Schuster, "Separation of Independent Sources Using Time-Delayed Correlations," *Physical Review Letters*, vol. 72, pp. 3634-3637, 1994.
- [6] Proakis G. John and M. G. Dimitris, *Digital Signal Processing: Principles, Algorithms and Applications*, Third ed. Upper Saddle River, New Jersey: Prentice-Hall, 1996.