IRWIN AND JOAN JACOBS
CENTER FOR COMMUNICATION AND INFORMATION TECHNOLOGIES

Blind Source Separation of Instantaneous Mixtures

Michael Shamis and Yehoshua Y. Zeevi

CCIT Report #787
March 2011

DEPARTMENT OF ELECTRICAL ENGINEERING
TECHNION - ISRAEL INSTITUTE OF TECHNOLOGY, HAIFA 32000, ISRAEL

CCIT Report #787 March 2011

Blind Source Separation of Instantaneous Mixtures

Michael Shamis Faculty of Electrical Engineering
Technion, Israel Institute of Technology
Haifa, Israel
Yehoshua Y. Zeevi Faculty of Electrical Engineering
Technion, Israel Institute of Technology
Haifa, Israel

***Abstract*—Blind source separation of images and voice signals is a well known and well studied subject. Solutions for this problem have various applications, such as separation of voices of multiple speakers in the same room, denoising, separation of reflections superimposed on images, and more.**

Classical time/position invariant Blind Source Separation is usually solved using Independent Component Analysis (ICA), which attempts to find statistically independent signals as a linear combination of the mixed signals, or by using Sparse Component Analysis (SCA) that estimates the mixing matrix by analyzing the geometry of the problem and uses scatter plots of the mixed signals to estimate co-linear centroids of the scattered data, where each centroid corresponds to a column of the mixing matrix.

Most of the studies in the field assume time/position invariant signal combinations, although many real life problems are not such. Recently, in his PhD thesis, Ran Kaftory has proposed an extension of the SCA method to solve multiple families of the time/position varying problems. He has shown that for instantaneous time/position varying mixtures, the problem of lines estimation transforms to estimation of nonlinear curves.

In this work we explore the separation of instantaneous time/position varying mixtures for which the parametric structure of the mixtures family is known apriori .We show that the geometric approach can also be viewed as Maximum Likelihood (ML) problem, when sparsification is applied to the mixed signals. We propose a multi-staged SCA algorithm for separation of time/position invariant mixtures and extend the solution to a subset of time/position varying mixtures where the reconstruction is performed by curve fitting techniques and nearest neighbor clustering.

In addition to the geometric approach, we extend the well-known technique of ICA by the ML approach, to the case of time/position varying instantaneous mixtures. We show that ML approach to the time/position varying separation problem can be developed from an information theoretic perspective as a joint Entropy minimization of the unmixed signals. We prove that although the problem is non-convex, and may require non-linear optimization techniques to solve, under certain conditions the correct signals separation constitutes a global maximum of the ML optimization problem.

We conclude by showing that the ML approach provides promising results, but due to the non-linear nature of the problem, its optimization is challenging and SCA-based approaches can be used as a complimentary technique to circumvent some of the difficulties originating from the non-linearities of the problem.

I. Introduction

Blind source separation (BSS) [12] has attracted a great deal of attention in recent decades. BSS deals with the decomposition of given mixtures of signals/images onto the original signals. This problem arises in real life applications in communication systems, voice processing, photography and filmography. The classical formulation of the problem is best illustrated by the Cocktail Party problem [12]. At Cocktail Party, a human brain is capable of separating single speaker from many and from additional background noises. Although we believe that human brain relies on additional information, such as visual information and prediction of words from the context, it is still possible to reconstruct the voice signal by using only multiple microphones/sensors which provide sufficient information for estimating the signals up to some error.

In this work we present results for both time/position varying and invariant mixtures but most of the focus of this work is on the BSS of time/position varying mixtures. The classical BSS approach assumes that the mixtures and weights do not change with time/position, however in real life application this assumption is almost never true. Even a simple Cocktail Party problem as described above is not time invariant due to the movement of the people and changes of the environment. Applications assuming that mixtures vary with time/position are much more complicated than those which assume the time/position invariance, due to the fact that the space of the reconstruction contains many degrees of freedom. In general, we believe that it is impossible to reconstruct signals from time/position varying mixtures without additional information or additional assumptions about the mixture families. In this work we assume that the mixing family is known up to several parameters. This assumption allows us to develop techniques for searching the optimal reconstruction parameters.

A. Problem Definition

1) Instantaneous Time/Position invariant BSS: Time/Position invariant BSS (TPIBSS) separation can be described as a Multiple Input-Multiple Output (MIMO) system, where the inputs are linearly mixed signals and the outputs are the reconstructions of the unmixed signals. The TPIBSS problem is a well studied problem and various solutions have been developed for the problem in the past decades. In many studies, the problem is solved using Maximum Likelihood (ML) or similar approach [2], [6], [10], [13], while other use geometric approach based on Sparse Component Analysis [4],[5],[25].

A formal definition of this problem assumes that there exists

a set of N signals and T samples of each signal

$$\{\{\mathbf{s}_1(\xi_1),...,\mathbf{s}_1(\xi_T)\},...,\{\mathbf{s}_N(\xi_1),...,\mathbf{s}_N(\xi_T)\}\}.$$

There also exists some unknown mixing matrix A of size $M \times N$. The samples are taken at a discrete time/postion grid $\xi_i \in \Xi$, where Ξ represents time for signals such as voice, and pixel locations for images. The input of the BSS system can then be described as a set of samples of M signals

$$\{\{\mathbf{x}_1(\xi_1),...,\mathbf{x}_1(\xi_T)\},...,\{\mathbf{x}_M(\xi_1),...,\mathbf{s}_M(\xi_T)\}\}$$

given by

$$\begin{pmatrix} \mathbf{x}_1(\xi) \\ \mathbf{x}_2(\xi) \\ ... \\ \mathbf{x}_M(\xi) \end{pmatrix} = A \begin{pmatrix} \mathbf{s}_1(\xi) \\ \mathbf{s}_2(\xi) \\ ... \\ \mathbf{s}_N(\xi) \end{pmatrix}. \quad (1)$$

The output of the TPIBSS system are the reconstructed samples of the signals $\{\{\mathbf{y}_1(\xi_1),...,\mathbf{y}_1(\xi_T)\},...,\{\mathbf{y}_N(\xi_1),...,\mathbf{y}_N(\xi_T)\}\}$, such that $\mathbf{y}(\xi) = \mathbf{s}(\xi)$ up to constant amplification factor and permutations. The amplification and permutation errors arise from the fact that it is impossible to identify whether these factors originated from the signals or from the mixing matrix which can be any general matrix.

The preceding problem is usually solved under the following assumptions [7], [12] :

- $N = M$, this assumption is used in all following sections unless explicitly mentioned otherwise.
- A is an invertible matrix
- The signals $\{\{\mathbf{s}_1(1),...,\mathbf{s}_1(T)\},...,\{\mathbf{s}_N(1),...,\mathbf{s}_N(T)\}\}$ are statistically independent (for each time/position sample).
- The signals $\{\{\mathbf{s}_1(1),...,\mathbf{s}_1(T)\},...,\{\mathbf{s}_N(1),...,\mathbf{s}_N(T)\}\}$ are non gaussian

Not all of the assumptions above are always necessary. The motivation for these assumptions and the cases in which some of them can be omitted will be explained in detail in later sections.

2) Instantaneous Time/Position Varying BSS: Instantaneous Time/Position Varying BSS (TPVBSS) is considered to be a difficult problem. To the best of our knowledge no solution exists for the general case of TPVBSS. Recently, solutions for special cases were presented in [1], [15], [24]; here we use the problem formulation that is based on [15]. The formulation of TPVBSS is very similar to that of Time/Position invariant. The only difference is that in the time/position varying case, we assume that the mixing matrix A is no longer constant, but varies with time/position. We also assume that the matrix A belongs to some known parametric family, and depends on an unknown vector of parameters θ of size K. The input signals $\{\{\mathbf{x}_1(\xi_1),...,\mathbf{x}_1(\xi_T)\},...,\{\mathbf{x}_M(\xi_1),...,\mathbf{x}_M(\xi_T)\}\}$ for the TPVBSS system are thus given by

$$\begin{pmatrix} \mathbf{x}_1(\xi) \\ \mathbf{x}_2(\xi) \\ ... \\ \mathbf{x}_M(\xi) \end{pmatrix} = A(\xi, \theta_{mix}) \begin{pmatrix} \mathbf{s}_1(\xi) \\ \mathbf{s}_2(\xi) \\ ... \\ \mathbf{s}_N(\xi) \end{pmatrix}, \quad (2)$$

where θ_{mix} is an unknown parameters vector that is used for creating the mixtures. The outputs of the system, as in the case of TPIBSS, are the reconstructed samples of signals $\{\{\mathbf{y}_1(\xi_1),...,\mathbf{y}_1(\xi_T)\},...,\{\mathbf{y}_N(\xi_1),...,\mathbf{y}_N(\xi_T)\}\}$ that should be equal to $\mathbf{s}(\xi)$ up to constant amplification factor and permutations. The solutions of this problem also require the same assumption as in the TPIBSS case.

Noted that the TPIBSS problem is a subclass of TPVBSS. As such, TPIBSS can easily be formulated as TPVBSS problem, where we assume that the parametric space of the mixing matrix is given by

$$A(\xi, \theta) \equiv \begin{pmatrix} \theta_1 & ... & \theta_{\mathbf{N}} \\ ... & ... & \\ \theta_{NM-M+1} & ... & \theta_{NM} \end{pmatrix}. \quad (3)$$

In other words, we constraint the mixing matrix to be in the space of all constant matrices.

B. Paper Contributions

- Extension of Maximum Likelihood BSS solution for TPVBSS:
  We developed an algorithm which is based on the ML solution of TPIBSS, that also solves various TPVBSS problems. We show that under certain conditions this algorithm provides optimal unmixing and analyze possible sources of errors that may lead to wrong results.
- Staged Sparse Component Analysis algorithm for TPIBSS:
  We developed a new algorithm that is based on existing Sparse Component Analysis techniques. It allows reconstruction of the mixing matrix step by step, overcoming difficulties encountered in estimating multiple maxima simultaneously.
- Staged Sparse Component Analysis algorithms for TPVBSS:
  We developed an algorithm which solves some of the TPVBSS cases using the Sparse Component Analysis approach, and presented comparison of this approach to the ML-based unmixing.

C. Paper Organization

In Section **??** we present the Maximum Likelihood approach for the solution of TPIBSS and extend the solution to the case of TPVBSS in Section III. Section **??** describes an alternative approach for solving both TPIBSS and TPVBSS problems that is based on image sparsification. We conclude by discussing the results of this work and recommendations in section VI.

II. Maximum Likelihood Blind Source Separation of Time/Position Invariant Mixtures

In this section we briefly review, as a background, commonly known Maximum Likelihood (ML) results [11], [12], [13]. We describe a gradient ascent [3] approach for ML, although better algorithms (such as fixed point and natural gradient algorithms) exist [9], [10], [14], we do not present them, since, they can not be extended for the solution of TPVBSS problem.

Assumptions needed for ML algorithm derivation for BSS:

- Source signals are independent and identically distributed for all ξ.
- Source signals joint probability distribution function is not defined by its first 2 moments.
- Probability density function $p_s(\mathbf{s}_i(i))$ of the original signals is known[1]. Note that although this assumption might not be correct, consistent estimator of the signals can still be achieved in some cases [21].

Under these assumptions, the probability density function of a vector of signals $\mathbf{s}(\xi)$ is given by

$$p(\mathbf{s}_1(\xi), \mathbf{s}_2(\xi), ..., \mathbf{s}_M(\xi)) = \prod_{i=1}^{M} p_s(\mathbf{s}_i(\xi)), \tag{4}$$

Let matrix B be the inverse of the mixing matrix A. It can be easily shown [12], that if the relation between given samples and the original signals is given by (1), then the probability distribution function of the given samples is

$$p(\mathbf{x}_1(\xi), \mathbf{x}_2(\xi), ..., \mathbf{x}_M(\xi)) = |\det B| \prod_{i=1}^{M} p_s(B_i^T \mathbf{x}_i(i)). \tag{5}$$

In order to find the probability of a set of samples, we perform quantization of the probability density function into small sized buckets. The probability that the set of mixture samples $\{\mathbf{x}_i(\xi)\}$ represents the signals mixed by the given matrix B^{-1} is:

$$\begin{aligned} F(B) &\equiv p(\mathbf{x}_1(\xi), \mathbf{x}_2(\xi), ..., \mathbf{x}_M(\xi)|B) = \\ &= \triangle^{MT} \prod_{\xi \in \Xi} |\det B| \prod_{i=1}^{M} p_s(B_i^T \mathbf{x}_i(\xi)), \end{aligned} \tag{6}$$

where $\triangle$ is an arbitrarily small-sized quantization bucket. Most of the studies in the field of BSS omit this term, since it does not affect the optimization problem, adding only a constant factor. We presented it here for correctness, but further more it's important to note that the quantization is performed in the space of the mixed signals, and not in the space of the source signals. Although for time/position invariant BSS it has no difference as will be shown later in this section, it has a serious impact on the time/position varying BSS.

Usually instead of maximizing the expression in (6), its log is maximized. The log likelihood is given by

$$\begin{aligned} L(B) &= \log F(B) = \\ &= T \log |\det B| + MT \log \triangle + \\ &\quad + \sum_{\xi \in \Xi} \sum_{i=1}^{M} \log p_s(B_i^T \mathbf{x}_i(\xi)). \end{aligned} \tag{7}$$

$L(B)$ is the log likelihood probability of the given samples. This expression is usually optimized for matrix B to achieve reconstruction. Note that although during the derivation of (7), we used the fact that the original signals are independent, the maximization of $L(B)$ does not guarantee that the reconstructed signals will be independent or uncorrelated. Thus the expression in (7) is usually optimized under the constraint that reconstructed signals are uncorrelated and have unit variance.

Lemma 2.1: In the maximization of the expression in (7) under the constraint that reconstructed signals are uncorrelated with unit variance, $|\det B|$ is a constant.

Proof: From the given constraint we can conclude that $R_{ss} = I$. However, since $\mathbf{s}(\xi) = B\mathbf{x}(\xi)$, it is known that $R_{ss} = BR_{xx}B^T$. Substituting the expressions for the covariance matrices and taking the determinant of both sides we get $\frac{1}{|\det R_{xx}|} = |\det B|^2$ implying that $|\det B|$ is a constant. ■

By omitting all the constant expressions from (7) we end up with a simpler target function

$$Q(B) = \sum_{\xi \in \Xi} \sum_{i=1}^{M} \log p_s(B_i^T \mathbf{x}_i(\xi)), \tag{8}$$

which can optimized under the constraint that the original signals are uncorrelated and have unit variance.

A. Alternative Approach for ML BSS of Time Invariant Mixtures

Instead of maximizing the probability that a set of given samples is generated by the mixing matrix B^{-1}, we can maximize the probability that reconstructed signals are indeed signals which match the predefined distribution. The main difference in the resulting formula arises from the fact that the quantization of the probability function is performed in the source signals space, instead of quantization in the mixed signal space. The probability in this case is

$$\hat{F}(B) = \triangle^{MT} \prod_{\xi \in \Xi} \prod_{i=1}^{M} p_s(B_i^T \mathbf{x}_i(\xi)), \tag{9}$$

The log likelihood of this expression is given by

$$\hat{L}(B) = MT \log(\triangle) + \sum_{\xi \in \Xi} \sum_{i=1}^{M} \log(p_s(B_i^T \mathbf{x}_i(\xi))), \tag{10}$$

Note that the expression in (10) is the same as in (8) up to a constant, thus we conclude that the ML algorithm for time/position invariant BSS is obtained from maximization of probability of mixed signals is the same as maximization of the probability of unmixed signals. This result is not trivial, since the quantization of the probability function does not affect the final probability formula in the same way in the above cases.

B. Predefined signals probability function

In the definition of the ML BSS solution we have assumed, so far, that the probability density function of the original signals is known apriori. In fact, in general applications this assumption is wrong and some approximation should be used. There exists a simple set of functions that provide surprisingly good reconstruction results for the BSS problems. One such function is:

$$f(v) \equiv \log p_s(v) \equiv \log \cosh(v) - \frac{v^2}{2} + a, \tag{11}$$

where the constant a does not affect the optimization and thus can be omitted. Note that without the constant a, $f(v)$

[1]This assumption is used for derivation of the algorithm and its justification will be shown later

is non positive for all v. This function represents a simple sub-Gaussian distribution and provides a good estimation for the probability distribution of images, which usually also have sub-Gaussian histograms [12].

C. Results

Here we show an example of the reconstruction using the maximization of the likelihood as described above. The expression in (10) is maximized under the constraint that the reconstructed signals are uncorrelated. This constraint is forced by first applying whitening and then performing the log likelihood maximization in the space of the orthonormal matrices [12]. Note that there exists more than one optimal reconstruction, since reconstruction of the negatives of the original signals is also an optimal reconstruction. In order to show the images properly they are normalized to be positive and have positive means. It also should be noted that some of the reconstructed images might seem lighter/darker due to the fact that the reconstruction can only be done up to a scaling factor. At Figure 2 we show 3 mixtures of signals that were generated with the following randomly generated mixing matrix

$$\begin{pmatrix} 4.9807 & 0.5333 & 3.8746 \\ 0.3909 & 4.8095 & 4.0865 \\ 2.2134 & 0.0232 & 4.3435 \end{pmatrix}, \tag{12}$$

from the original signals at Figure 1. At Figure 3 it can be seen that the reconstructed signals are very similar to the original ones up to scaling and permutation.

Fig. 1. Original images

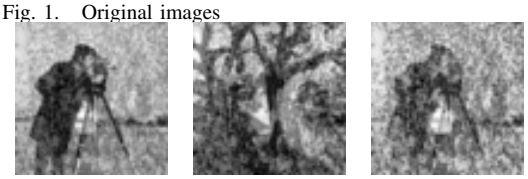

Fig. 2. Mixed images

Fig. 3. Reconstructed images

D. Entropy and Mutual Information Minimization

Another useful approach which leads to the same result as in the previous sections is the minimization of mutual information between reconstructed signals. It is a well known result [8] that Mutual Information of a set of random variables achieves its minimal value of 0 only for an independent set of random variables (for non singular distributions). From Lemma. II we can see, that the determinant of matrix B is constant, thus for every reconstruction matrix B, the mutual information of the reconstructed signals is given by

$$\begin{aligned} I(B\mathbf{x}(\xi)) &= \sum_{i=1}^{N} H(B_i^T \mathbf{x}_i(\xi)) - H(B\mathbf{x}(\xi)) = \\ &= \sum_{i=1}^{N} H(B_i^T \mathbf{x}_i(\xi)) - H(\mathbf{x}(\xi)) - E\{\log|\det B|\} \end{aligned} \tag{13}$$

The last two expression in (13) are constant for any matrix B during the reconstruction, thus the minimization of mutual information of the signals boils down to the minimization of sums of entropies of the reconstructed signals, which leads to the same solution as in the case of Negentropy maximization and the ML approach.

III. Maximum Likelihood Blind Source Separation of Time/Position Varying Mixtures

When considering the time/position varying BSS problem, we assume that the input signals to the unmixing system are generated by the model described by (2). We also assume that the parametric model of the mixing matrix $A(\xi, \theta)$ is known - and thus the parametric family of the inverse matrices $B(\xi, \theta)$ is also known.

A. Naive Reconstruction

A naive extension of the method proposed in Section II can be derived, by starting from the assumption that an approximation of probability function of the original signals is known and given by (11). Thus the probability distribution of input signals of the system for a specific time/space point is given by

$$p(\mathbf{x}_1(\xi), \mathbf{x}_2(\xi), ..., \mathbf{x}_M(\xi)) = |\det B(\xi,\theta)| \prod_{i=1}^{M} p_s(B_i^T(\xi,\theta)\mathbf{x}_i(\xi)). \tag{14}$$

Once again we shall use the approximation of the discrete probability function to measure the probability of obtaining the set of given samples. The probability that a given set of input samples generated with the given probability function and with the given θ is then given by

$$F(\theta) = \triangle^{MT} \prod_{\xi\in\Xi} |\det B(\xi,\theta)| \prod_{i=1}^{M} p_s(B_i^T(\xi,\theta)\mathbf{x}_i(\xi)). \tag{15}$$

Instead of maximizing the function F for the parameters vector θ, we optimize its log

$$\begin{aligned} L(\theta) &= MT\log(\triangle) + \sum_{\xi\in\Xi} |\log|\det B(\xi,\theta)| + \\ &+ \sum_{\xi\in\Xi}\sum_{i=1}^{M} \log(p_s(B_i^T(\xi,\theta)\mathbf{x}_i(\xi))). \end{aligned} \tag{16}$$

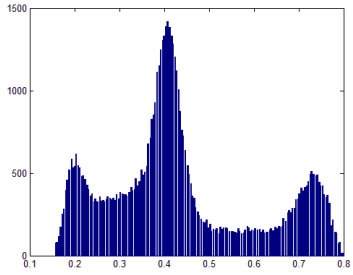

Fig. 4. Non normalized cameraman histogram

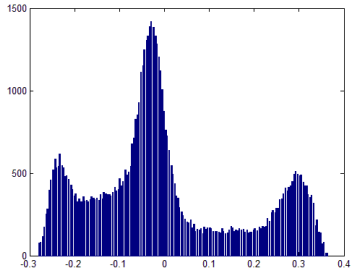

Fig. 5. Histogram of cameraman with zero mean

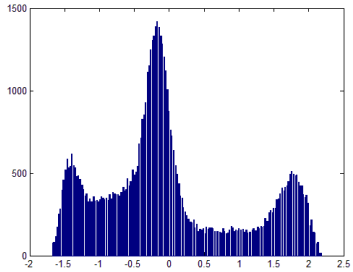

Fig. 6. Histogram of cameraman with zero mean and unit variance

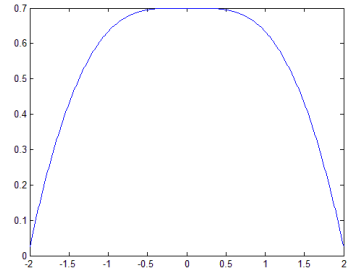

Fig. 7. Estimation of the histogram using (11)

One could expect that optimizing the function in (16) with respect to θ, under the constraint that reconstructed signals should be uncorrelated would provide a good reconstruction of the original signals. In practice, however, this approach does not perform well on most of the parametric families for various reasons:

- One of the main problems of the naive approach arises from the fact that the approximation of a probability function p_s is a good approximation only for signals with zero mean and unit variance. For example, it can be seen that the estimation of the histogram at Figure 7 is a much better estimation for the normalized histogram of the cameraman image (Figure 6) than the histogram of the non normalized cameraman image (Figure 4) or of the cameraman image with just the mean value normalizated (Figure 5). Note that although the quality of the approximation can be seen by eye, we can define the quality of approximation using KL divergence as we will show later. For the time/position invariant case, the problem of non normalized signals is solved by applying whitening onto the input signals and then maximizing the probability in the space of unitary reconstruction matrices - it preserves the invariant of the zero mean and the unit variance of the reconstructed signal. In the TPIBSS case whitening is not an option, since after whitening it is very likely that there does not exist θ that reconstructs the signals (whitening changes the family of the parametric mixtures, since it adds additional linear mixing). For general parametric families there exist no preprocessing of the signals that would assure unit variance and zero mean invariant. We solve this problem in the next sections by defining regularization of the reconstructed signals at each step of the algorithm to preserve those statistical invariants for the reconstructed signals.
- An additional problem with the naive approach is that for many parametric families the expression $\sum_{\xi\in\Xi} |\log(|\det B(\xi,\theta)|$ is unbounded for the parameters vector θ and the maximization of the expression in (16) usually does not converge, since there are paths that lead to infinite increase of this expression - usually this would be the dominant element in the log likelihood expression. The reason that this problem does not occur in the time/position invariant case can be explained by Lemma II which proves that for the TPIBSS the analogue of this expression is a constant, but this Lemma is no longer valid for the TPVBSS problem and thus this element causes divergence of the solution. Careful analysis of the sources of this expression reveals that it appears due to the fact that quantization of the probability function is performed in the space of the input signals. The problem can be resolved by taking the alternative approach as described in Section II-A.

B. Reconstruction of Signals in The Space of Normalized Signals

The approach in this section is based on the naive reconstruction that was described in Section III-A. The problems of the naive approach are solved by performing a normalization of the reconstructed signals. For this purpose we define the vector of the non normalized reconstructed signals for the given θ as

$$\begin{pmatrix} \mathbf{z}_1(\xi,\theta) \\ \mathbf{z}_2(\xi,\theta) \\ \dots \\ \mathbf{z}_M(\xi,\theta) \end{pmatrix} = B(\xi,\theta) \begin{pmatrix} \mathbf{x}_1(\xi) \\ \mathbf{x}_2(\xi) \\ \dots \\ \mathbf{x}_M(\xi) \end{pmatrix}. \qquad (17)$$

The mean vector of these signals for the given θ is

$$\mathbf{m}(\theta) \equiv \frac{1}{T} \sum_{\xi \in \Xi} \mathbf{z}(\xi, \theta). \tag{18}$$

We can now define the variance normalization matrix as

$$N(\theta) = \mathrm{diag} \begin{pmatrix} \sqrt{\frac{T-1}{\sum_{\xi \in \Xi} \mathbf{z}_1(\xi,\theta)^2 - T\mathbf{m}_1(\theta)^2}} \\ \sqrt{\frac{T-1}{\sum_{\xi \in \Xi} \mathbf{z}_2(\xi,\theta)^2 - T\mathbf{m}_2(\theta)^2}} \\ . \\ . \\ \sqrt{\frac{T-1}{\sum_{\xi \in \Xi} \mathbf{z}_M(\xi,\theta)^2 - T\mathbf{m}_M(\theta)^2}} \end{pmatrix}. \tag{19}$$

The normalized reconstructed signals are now given by

$$\mathbf{y}(\xi, \theta) = N(\theta) \left(\mathbf{z}(\xi, \theta) - \mathbf{m}(\theta) \right) \tag{20}$$

Lemma 3.1: The normalized signals vector $\mathbf{y}(\xi, \theta)$ has zero mean[2] (with respect to ξ).

Lemma 3.2: Each normalized signal $\mathbf{y}_i(\xi, \theta)$ has unit variance.

From the results above we can conclude that the signals vector $\mathbf{y}(\xi, \theta)$ is a normalized estimation of the reconstructed signals for the given θ. As in the case of TPIBSS we can now define the log likelihood for a given set of normalized reconstructed samples for each θ as

$$L(\theta) = \sum_{\xi \in \Xi} \sum_{i=1}^{M} \log(p_s(\mathbf{y}_i(\xi))). \tag{21}$$

Once again it is useful to note that the expression in (21) can be seen as an estimation of the sum of the negation of the entropies of the reconstructed signals, where p_s is assumed to be the probability function of the signals. Note that from the fact that $\log(p_s(v))$ is non positive expression, follows that $L(\theta)$ is also non positive for all θ.

[2]Note that all claims about statistical measures are actually claims about their estimates from the signals

Fig. 8. Original images

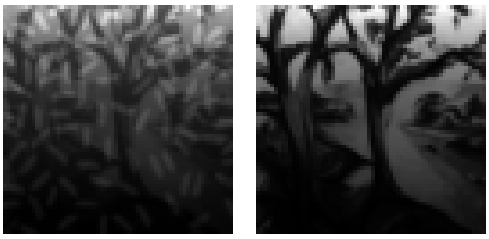

Fig. 9. Mixed images

Fig. 10. Reconstructed images with no regularization applied

Maximization of the expression in (21) provides good reconstruction results for many parametric families and solves the problems which were an inherent part of the naive approach. However, careful analysis of the proposed approach reveals that it does not constraint the reconstructed signals to be independent. In fact, for some parametric families all the reconstructed signals can be the same. As a simple example of such a case, we can look at the parametric family for which the reconstruction matrix is given by

$$B(\xi, (\theta)) = \begin{pmatrix} \theta\xi & (1-\theta)\xi \\ 0 & \xi \end{pmatrix}, \tag{22}$$

for this parametric family, the same signal will be reconstructed twice for $\theta = 0$ and if this signal has a large Entropy, then the global maximum will be close to it. Figure 11 illustrates the energy graph for various values of θ. The plot is presented for images that were mixed with $\theta = 2$, however, as it can be seen the maximal energy value is achieved at $\theta = 0$. This happens because the energy function as defined in (21) does not penalize for reconstruction of correlated signals. As can be seen at Figure 10 the reconstruction is this case is very poor, and only one signal is reconstructed.

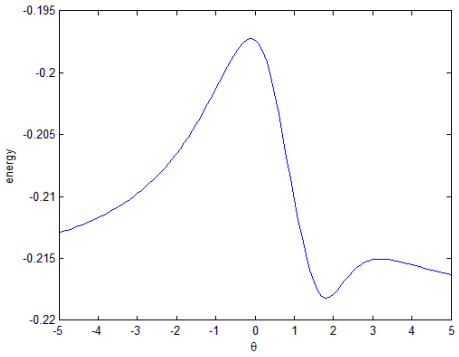

Fig. 11. Energy of a mixture of a two images, the mixing parameter is $\theta = 2$. The point $\theta = 2$ is not even local maximum due to the fact the reconstructed signals are not constrained to be uncorrelated

One way of solving this problem is performing constrained optimization of the energy function from (21). We do not use this approach due to the fact that under the constraint of uncorrelated signals reconstruction, the function can have multiple local maxima. With multiple restarts and simulated annealing [20], the constrained optimization problem is very expensive computationally. As an alternative for this approach we present a regularization penalty factor based on the reconstructed signals covariance matrix $R_{yy}(\theta)$ defined by

$$R_{yy}(\theta) \equiv E\left\{\mathbf{y}(\xi,\theta)\mathbf{y}^T(\xi,\theta)\right\}. \tag{23}$$

Let us define the penalty factor as:

$$P(\theta) \equiv \log|\det R_{yy}(\theta)| \tag{24}$$

Lemma 3.3: $P(\theta)$ achieves its minimal value of 0 when the reconstructed signals are uncorrelated.

Proof: From the definition of the correlation matrix, if the reconstructed signals $\mathbf{y}(\xi,\theta)$ are uncorrelated, then $R_{yy}(\theta)$ is diagonal. The elements on the diagonal are exactly the variance of each signal. From Lemma 3.2 all the variances are 1, thus for uncorrelated signals reconstruction $R_{yy}(\theta) = I$ and thus its determinant equals to 1 and the its log is equal to 0. ■

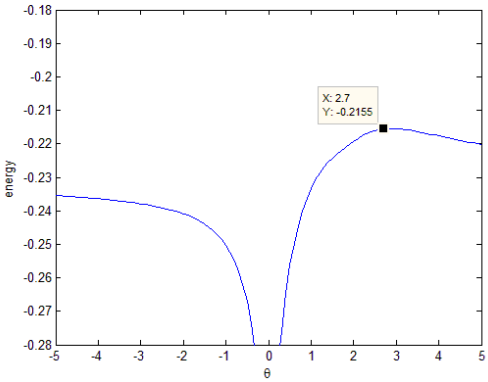

Fig. 12. The energy of a mixture of a two images, the mixing parameter is $\theta = 2$. The regularization element is applied with $\lambda = 2$, the global maximum is now located at $\theta = 2.7$ which provides much better reconstruction than $\theta = 0$.

Using the penalty factor we now define the energy function which penalizes the reconstruction of correlated signals as

$$\begin{aligned} F(\theta) &= L(\theta)\left(1+\lambda P(\theta)\right) = \\ &= \left(\sum_{\xi\in\Xi}\sum_{i=1}^{M}\log(p_s(\mathbf{y}_i(\xi)))\right) + \\ &+ \left(\sum_{\xi\in\Xi}\sum_{i=1}^{M}\log(p_s(\mathbf{y}_i(\xi)))\right)\log|\det R_{yy}(\theta)| \end{aligned} \tag{25}$$

The penalty function is multiplicative and non additive in order to normalize units. The penalty function values are usually larger than the values $L(\theta)$. Therefore, if we would just add the penalty function, then the problem would become very similar to decorrelation which is of course an unwanted result. It should be noted that the regularization factor $\lambda L(\theta)P(\theta)$ is always non-positive due to the fact that $L(\theta)$ is non positive and $P(\theta)$ is positive for all θ. Thus the regularization factor achieves its maximal value of 0 for uncorrelated reconstruction signals. It can be seen on Figure 12, that when the regularization is applied for the same example as in Figure 11 the maximum is now achieved at 2.7 which is closer to the real value of the mixing parameter 2, it can also be seen that the reconstructed signals for this parameter value are now similar to the original ones (Figure 15).

Fig. 13. Original images

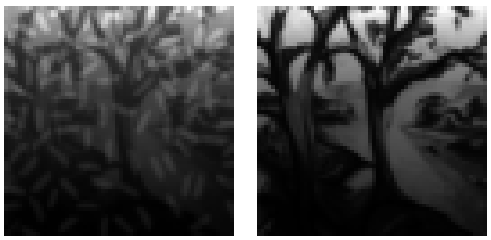

Fig. 14. Mixed images

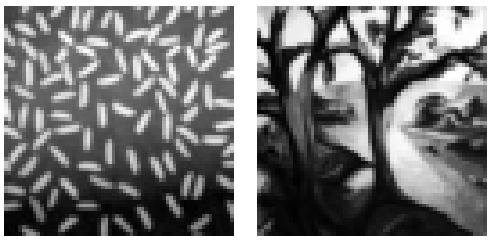

Fig. 15. Reconstructed images when regularization is applied with $\lambda = 0.1$

C. Reconstruction in the Space of Normalized Signals Using Mutual Information Minimization

As mentioned above, one of the interpretations of the ML reconstruction in the time/position invariant case is the

minimization of Mutual Information. It allows development of similar reconstruction algorithm, which under certain conditions can be proven to be optimal. Thus, we can define our goal as minimization of the mutual information of the normalized reconstructed signals (which are defined in (20)):

$$\begin{aligned} I(\mathbf{y}(\xi,\theta)) &= \sum_{i=1}^{M} H(\mathbf{y}_i(\xi,\theta)) - H(\mathbf{y}(\xi,\theta)) = \\ &= \sum_{i=1}^{M} H(\mathbf{y}_i(\xi,\theta)) - H(\mathbf{x}(\xi)) - \\ &\quad -E\left\{\log|\det N(\theta)B(\xi,\theta)|\right\} = \\ &= \sum_{i=1}^{M} H(\mathbf{y}_i(\xi,\theta)) - H(\mathbf{x}(\xi)) - \\ &\quad -E\left\{\log||\det B(\xi,\theta)|\right\} - \\ &\quad -\log|\det N(\theta)|. \end{aligned} \tag{26}$$

Lemma 3.4: If the original signals $\mathbf{s}(\xi)$ are iid, then θ_{mix} that reconstructs these signals (up to a multiplicative constant) is a global minimum of the expression in (26).

Proof: The minimal possible value of the mutual information is 0 and it is achieved for the reconstruction of independent signals. θ_{mix} achieves the reconstruction of independent signals and thus $I(\mathbf{y}(\xi,\theta_{mix}))=0$ is a global minimum (not necessarily unique). ∎

The expression $H(\mathbf{x}(\xi))$ is constant - it is the Entropy of the original mixtures. Thus the minimization of the mutual information can be written as the maximization of

$$E\left\{\log|\det B(\xi,\theta)|\right\} + \log|\det N(\theta)| - \sum_{i=1}^{M} H(\mathbf{y}_i(\xi,\theta)). \tag{27}$$

Once again we assume that the pdf of the reconstructed signals is known and is equal to p_s for all reconstructed signals. Thus an estimate of the sum of entropies from the samples of the reconstructed signals is given by

$$\begin{aligned} -\sum_{i=1}^{M} H(\mathbf{y}_i(\xi,\theta)) &= \sum_{i=1}^{M} E\left\{\log(p_s)\right\} \approx \\ &\approx \frac{1}{T}\sum_{i=1}^{M}\sum_{\xi\in\Xi}\log(p_s(\mathbf{y}_i(\xi))). \end{aligned} \tag{28}$$

The expression that should be maximized is given by

$$\begin{aligned} F_I(\theta) &= \frac{1}{T}\sum_{i=1}^{M}\sum_{\xi\in\Xi}\log(p_s(\mathbf{y}_i(\xi))) + \\ &\quad +\frac{1}{T}\sum_{\xi\in\Xi}\log|\det B(\xi,\theta)| + \log||\det N(\theta)| \end{aligned} \tag{29}$$

Note that the expression in (29) does not require any additional regularization since reconstruction of the same signal more than once can never yield the maximum due to the fact that $\sum_{\xi\in\Xi}\log|\det B(\xi,\theta)|$ will be equal to minus infinity for such reconstruction.

Lemma 3.5: If the original signals $\mathbf{s}(\xi)$ are iid with unit variance, the pdf of the all of the original signals(normalized to unit variance and zero mean) is equal to p_s and the number of signal samples tends to infinity then the real mixing parameter $\theta_{\mathbf{mix}}$ is the global maximum of the expression in $F_I(\theta)$.

Proof: When the number of samples tends to infinity then $F_I(\theta)$ tends to negation of the expression in (26) (up to a constant), since all the statistical estimates converge to their real values (the Entropy estimation is also correct since we assumed that the pdf estimation is perfect). Thus, $F_I(\theta)$ achieves its global maximum for the same value of θ as (26), and from Lemma 3.4 it is achieved for a perfect reconstruction of θ_{mix}.

∎

D. Discussion

The approach described in this section, works well on numerous parametric families, but still there exist parametric families for which it would not work well. Here we try to better understand when does the global maximum of the expression in (25), (29) indeed corresponds to a good reconstruction.

Some parametric families vary significantly with time. As a simple example of such a parametric family we can consider a family with the reconstruction matrix

$$B(\xi,\theta) = \begin{pmatrix} 1 & \theta_1||\xi||^3 \\ \theta_1||\xi||^3 & 1 \end{pmatrix}. \tag{30}$$

Since the reconstruction approach relies on statistics, it is desirable that the mixtures will not change rapidly in time/position. For parametric families, such as those of (30), the elements for larger values of $||\xi||$ become much more significant than the samples for smaller values of $||\xi||$. This causes the covariance matrix estimation in the regularization term of (25) to depend primarily on the last samples (samples of large $||\xi||$). In such cases the covariance estimation causes invalid behavior of the regularization factor. This issue is partially solved, using the predefined probability density function, which penalizes such singular distribution functions. However the algorithm will fail to converge in most of such cases.

Another issue that may affect correctness of the reconstruction is the fact that the Entropy used in the optimization problem is estimated using a simple probability function. For each random variable y this approximation introduces an estimation error that can be formulated as follows

$$\begin{aligned} H_y - E\{-\log p_s\} &= H_y + \int p_y(v)\log(p_s(v))dv = \\ &= H_y - \int p_y(v)\log\left(\frac{1}{p_s(v)}\right)dv = \\ &= H_y - \int p_y(v)\log\left(\frac{p_y(v)}{p_s(v)}\right)dv + \\ &\quad + \int p_y(v)\log(p_y(v))dv = \\ &= H_y - D_{KL}(p_y||p_s) - H_y = \\ &= -D_{KL}(p_y||p_s). \end{aligned} \tag{31}$$

Thus the introduced error equals to minus the KL divergence between the probability function of random variable and its estimate. The error factor is always negative and is zero if and only if the estimated probability density function coincides with the probability density function itself. From this it

becomes clear, that if there exist θ_2 for which p_s constitutes a better approximation of the probability function, than for the original signal probability function, it can cause the appearance of an invalid global maximum. Note that this error function is highly dependent on the probability distribution of the original signals, and thus no explicit expression for the error factor can be derived.

E. Results

In this section we show results of the reconstruction using the maximization of the expressions in (25) and (29). The maximization was performed using a gradient ascent algorithm with golden section algorithm for finding maximum in the specific direction [3]. In addition we used a multiple restart strategy and simulated annealing [20] in order to avoid local maxima. The high level algorithm applied for the reconstruction is given at Algorithm 1. Note that due to the non-deterministic nature of the algorithm, for some runs it does not converge to the correct reconstruction vector. However if a sufficient number of iterations and restarts is used, the probability for this tends to 0. It should also be noted that for all parametric families on which we tested the algorithm, optimization of (29) always provided similar or better results than those that were achieved by optimizing (25).

Algorithm 1 Calculate $\theta_{opt} = R(mixedImages, maxIter, maxRest, maxTemp)$

```
numRestarts ← 0
θ_opt ← randomInitValue
energy_opt ← F(θ_opt)
while numRestarts ≤ maxRestarts do
  θ ← randomInitValue
  i ← 0
  while i ≤ maxIter do
    T = maxTemp (1 − i/maxIter)
    randMove = makeRandomMove(T)
    if shouldMakeRandomMove(T, F(θ), F(θ + randMove)) then
      θ ← θ + radnomMove
    else
      gradF ← ∇F(θ)
      bestStep ← argmax_α F(θ + α · gradF)
      θ ← θ + α · gradF
    end if
    if energy_opt < F(θ) then
      θ_opt ← θ
      energy_opt ← F(θ_opt)
    end if
  end while
end while
```

Figure 18 illustrates the reconstruction results for the maximization of (25) for a parametric family given by

$$A(\theta, c) = \begin{pmatrix} 1 & 1+\theta_1 c \\ 1+\theta_2 c & 1 \end{pmatrix} \tag{32}$$

where c is proportional to the column number. As can be seen, the reconstruction provides good, although non perfect, visual results in this case. The reconstruction using (29) provides almost similar results and thus we omit it here.

Fig. 16. Original images

Fig. 17. Mixed images

Fig. 18. Reconstructed images when regularization is applied with $\lambda = 0.1$

On the other hand, for a simple parametric family, given by a rotation matrix (varying with parameter r that is proportional to the row index)

$$A(\theta, r) = \begin{pmatrix} cos(\theta_1 r) & sin(\theta_1 r) \\ -sin(\theta_1 r) & cos(\theta_1 r) \end{pmatrix} \tag{33}$$

the reconstruction result is incorrect for the maximization of (25) as seen in Figure 21. The optimization of the same example using the expression in (29) provides good reconstruction results as can be seen in Figure 22.

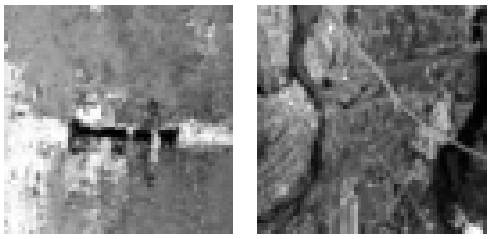

Fig. 19. Original images

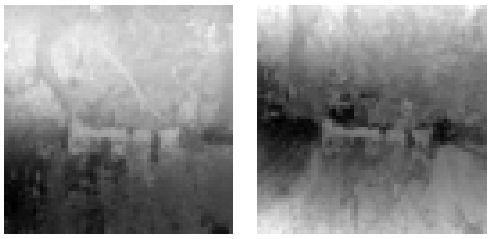

Fig. 20. Mixed images

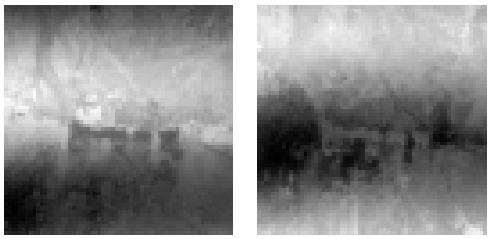

Fig. 21. Reconstructed images when regularization is applied with $\lambda = 0.1$

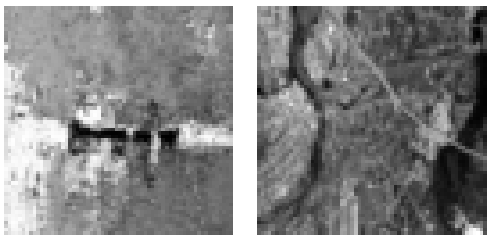

Fig. 22. Reconstructed images using maximization of (29)

IV. SSCA Separation of Time/Position Invariant Mixtures

In this section we present the application of Staged Sparse Component Analysis (SSCA) in solving the BSS problem [15], [16], [17], [18], [19]. We offer here a different approach here and extend it to perform on a larger set of parametric families. In this Section we present the approach for time/position invariant mixtures and later we extend it to time/position varying mixtures in Section V.

A. Sparsification

As follows from the title, SSCA is based on the sparse representation of the signals/images at hand. We assume that there exists a transformation from the space of source signals to the space of sparse signals (usually nonlinear), that can be applied to each of the mixed signals[3], $\Psi[\mathbf{x}](\xi)$, and has the following properties:

[3]When we use the notation of a transformation to a vector of signals, it means that the transformation is applied on each of the signals separately and the result is a vector of the transformed signals

1) At most one signal is active at any given time/position: for each ξ, the probability that $\Psi[\mathbf{s}](\xi)$ is comprised of more than one signal larger than zero is significantly smaller than the probability that only one signal is larger than zero.
2) Invariance to a mixing process: $\Psi[\mathbf{x}](\xi) \approx A\Psi[\mathbf{s}](\xi)$. This means that the order of applying the transformation and the mixing process does not affect significantly the result.
3) Zero mean: $E\{\Psi[\mathbf{x}](\xi)\} = 0$

In order to satisfy these constraints we should restrict ourselves to the family of sparse transformations that depend on a parameter δ, for which there exists a continuous function $k(\delta)$, and $\delta_0 \geq 0$, such that for all signals under consideration the following holds:

$$p(|\Psi_\delta[\mathbf{s}_i](\xi)| > 0) \leq k(\delta) \quad \forall i, \xi, \delta \tag{34}$$
$$p(|\Psi_\delta[\mathbf{s}_i](\xi)| > \delta) > 0 \quad \forall i, \xi, \delta > \delta_0 \tag{35}$$
$$p(|\Psi_\delta[\mathbf{s}_i](\xi)| > 0) = 0 \quad \forall i, \xi, \delta_0 \geq \delta \geq 0 \tag{36}$$
$$k(\delta_1) \leq k(\delta_2) \Leftrightarrow \delta_1 \leq \delta_2. \tag{37}$$

Let us denote by $\tilde{p}_\delta(m)$ the probability that exactly m signals have value larger than 0. Then if follows immediately from Lemma.4.1 that every such transformation satisfies property 1. Thus, we should look for the transformation that satisfies the invariance to the mixing process.

Lemma 4.1: Every transformation $\Psi_\delta[s]$ that has a function $k(\delta)$ that satisfies (34)-(37) satisfies property 1 - $\lim_{\delta\to\delta_0} \sum_{m=2}^{M} \frac{\tilde{p}_\delta(m)}{\tilde{p}_\delta(1)} = 0$, for a set of M independent stationary signals $\mathbf{s}(\xi)$ that have the same probability distribution.

Proof: Let us define

$$p_\delta \equiv p(|\Psi_\delta[\mathbf{s}_i](\xi)| > 0). \tag{38}$$

It is well defined and independent of i, since we assume that all signals have the same probability distribution. The signals are independent and, thus, their transformation must also be independent. The probability that exactly m signals have value greater than 0 is given by [4]

$$\tilde{p}_{\delta,\epsilon} = \frac{M!}{m!M-m!} p_\delta^m (1-p_\delta)^{M-m}. \tag{39}$$

Since $\lim_{\delta\to\delta_0} p_\delta = 0$, for small enough δ the following holds for $m \geq 2$:

$$m!(1-p_\delta)^{m-1} \geq 1 \tag{40}$$

and, thus, for all $m \geq 2$:

$$\frac{\tilde{p}_\delta(m)}{\tilde{p}_\delta(1)} = \frac{\frac{M!}{m!M-m!} p_\delta^m (1-p_\delta)^{M-m}}{\frac{M!}{M-1!} p_\delta (1-p_\delta)^{M-1}} = \tag{41}$$
$$= \frac{M-1!}{m!M-m!} \frac{p_\delta^{m-1}}{(1-p_\delta)^{m-1}} \leq \tag{42}$$
$$\leq \frac{M-1!}{M-m!} p_\delta^{m-1} \leq \frac{M-1!}{M-m!} k(\delta)^{m-1}. \tag{43}$$

[4]This expresses merely the probability that in a set of M binomial experiments there are m times success

Since M is a finite number and $\lim_{\delta\to\delta_0} k(\delta) = 0$, it immediately follows that

$$\lim_{\delta\to\delta_0} \frac{\tilde{p}_\delta(m)}{\tilde{p}_\delta(1)} = \lim_{\delta\to 0} \frac{M-1!}{M-m!} k(\delta)^{m-1} = 0 \tag{44}$$

and also

$$\lim_{\delta\to\delta_0} \sum_{m=2}^{M} \frac{\tilde{p}_\delta(m)}{\tilde{p}_\delta(1)} = 0. \tag{45}$$

■

In this work we focus on sparsification using wavelet transforms. This is done by performing wavelet packet decomposition of the images for 2 levels, and then for each pixel of the image we define $\Phi\{\mathbf{s}_i\}(\xi)$ as the closest (spatially) value at one of the high frequency bands. Then, using the threshold function

$$u_\delta(v) = \begin{cases} v & v \geq \frac{1}{\delta} \\ 0 & else \end{cases}, \tag{46}$$

we can define the transformation

$$\Psi_\delta[\mathbf{s}_i](\xi) = u_\delta(\Phi\{\mathbf{s}_i\}(\xi)). \tag{47}$$

For most of the images, the number of pixels that contain high frequencies is small. Therefore for small enough values of δ the above transformation indeed serves the purpose of sparsification. Furthermore, it can be easily seen that the mean value of the transformed signals is zero, since highpass filtering is used.

The sparsification is almost invariant to linear transformations, since wavelet transform is invariant to linear transformations and the only factor that affects this invariance is the nonlinearity u_δ. The mixing coefficients can cause significant change in the energy of the mixed signal (and thus in the energies of the channels). Thus values that did not pass the threshold before the linear transformation, will pass it after the transformation and vice-versa.

B. Separation of Mixtures

As mentioned in section I, the reconstruction of the signals can be accomplished up to a constant scaling factor. In the ML approach we reconstructed the signals with unit variance, whereas here we take a different approach and reconstruct the signals with the same magnitude as in the first mixture. We assume that all the coefficients of the mixing matrix are non zero for the first row, and thus we can define an alternative mixing matrix whose elements are defined by

$$\hat{A}_{i,j} = \frac{A_{i,j}}{A_{1,j}}. \tag{48}$$

Assuming that the mixing matrix is $\hat{A}$ instead of A, we should reconstruct the signals $\hat{\mathbf{s}}_i(\xi) = A_{1,i}\mathbf{s}_i(\xi)$. Note that if the original signals $\mathbf{s}_i(\xi)$ are independent, so are the signals $\hat{\mathbf{s}}_i(\xi)$.

Let us define $\Xi_L = \{\xi_1, \xi_2, ..., \xi_L\}$ as a set of L values for which

$$\Psi_\delta[\mathbf{x}_1](\xi) \neq 0 \quad \xi \in \Xi_L, \tag{49}$$

where Ψ_δ is a sparsification that satisfies the conditions stated in Section IV-A. For all $\xi \in \Xi_L$ we obtain

$$\frac{\Psi_\delta[\mathbf{x}_i](\xi)}{\Psi_\delta[\mathbf{x}_1](\xi)} = \frac{\Psi_\delta\left[\hat{A}_i^T\hat{\mathbf{s}}\right](\xi)}{\Psi_\delta[\mathbf{1}^T\hat{\mathbf{s}}](\xi)} \approx \frac{\hat{A}_i^T\Psi_\delta[\hat{\mathbf{s}}](\xi)}{\mathbf{1}^T\Psi_\delta[\hat{\mathbf{s}}](\xi)} \tag{50}$$

and, according to Lemma 4.1, we may conclude that with high probability only one signal has non-zero value and thus for each ξ it is useful to define

$$r_i(\xi) \equiv \frac{\Psi_\delta[\mathbf{x}_{i+1}](\xi)}{\Psi_\delta[\mathbf{x}_1](\xi)} \quad i \in 1..M-1. \tag{51}$$

Treating $r_i(\xi)$ as a random variable, we can see that ideally the probability distribution $p_{r_i}(v)$ should be a sum of delta functions, each one corresponding to a different coefficients of the matrix $\hat{A}_{i+1,j}$.

1) Reconstruction in case of 2 signals: As mentioned above, in an ideal case, we would have one or two delta functions as a probability distribution function of r_1. In practice, however, $r_1(\xi)$ is usually noisy and no explicit delta functions exist. This happens since the sparsification is not ideal and in most cases every value of the sparsified mixture signals is a sum of values contributed by multiple source signals (where only one value is large and all the other values are significantly smaller). Figure 23 illustrates an example plot of the histogram of $r_1(\xi)$ for 2 images with a mixing matrix

$$\hat{A} = \begin{pmatrix} 1 & 1 \\ 2 & 6 \end{pmatrix}, \tag{52}$$

where the threshold $\frac{1}{\delta}$ was chosen such that approximately 2% of the pixels of the mixture corresponding to the first row would pass. The histogram was calculated by quantizing each value to a bucket of size[5] 0.1. The two maxima are around the values 2 and 6, but instead of punctuate distribution, there are clouds of the values around the two maxima.

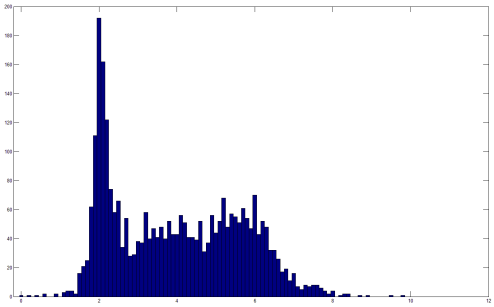

Fig. 23. Histogram of $r_1(\xi)$ for mixture of two signals, where $\hat{A}_2 = (2, 6)$

Using the histogram as a probability function for the means of finding maxima is challenging due to its non continuous nature. A smoother estimation of $p_{r_1}(\xi)$ can be achieved by using a kernel approximation for the probability distribution. In this work we use a Gaussian kernel, defined as

$$K_\sigma(v) \equiv \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{v^2}{2\sigma^2}}. \tag{53}$$

[5]The term "bucket size" is used here to specify the quantization resolution

Using this kernel, the approximation of $p_{r_1}(\xi)$ is given by

$$p_{r_1}(v) = \frac{1}{L} \sum_{\xi \in \Xi_L} K_\sigma(r_1(\xi) - v). \quad (54)$$

The smooth version of the probability function is depicted in Figure 24, for $\sigma = 0.5$. The maxima of the probability function appear close to 2 and 6.

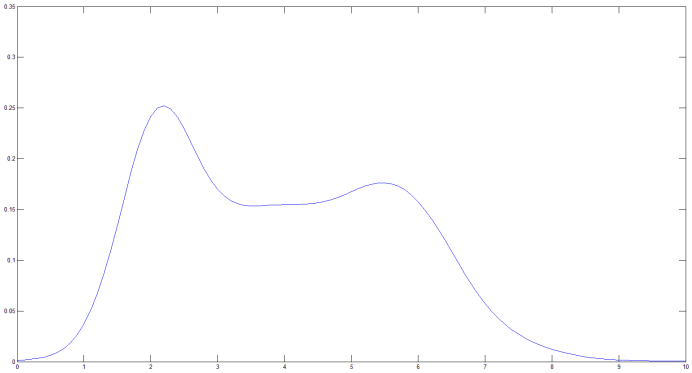

Fig. 24. The probability function p_{r_1} for mixture of two signals, estimated with kernel $K_{0.5}(v)$ for $\hat{A}_2 = (2, 6)$

These maxima can now be found by sampling the space of possible values (it is a one-dimensional, usually bounded, space), by using a gradient ascent algorithm with multiple restart points, or by using clustering algorithms such as K-Means, where each cluster will match a different coefficient in $\hat{A}_2$. An additional algorithm for finding the maxima is presented later in this section.

Given the two maxima that correspond to the matrix coefficients $\hat{A}_{2,1}, \hat{A}_{2,2}$, the estimation of the source signals is easily obtained by

$$\tilde{\mathbf{s}}(\xi) = \begin{pmatrix} 1 & 1 \\ \hat{A}_{2,1} & \hat{A}_{2,2} \end{pmatrix}^{-1} \mathbf{x}(\xi) \quad (55)$$

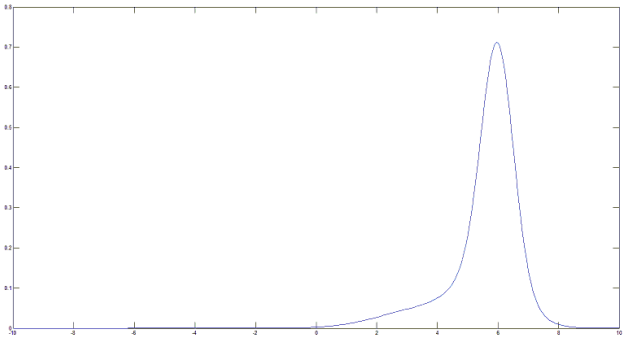

Fig. 25. p_{r_1} for mixture of two signals estimated with kernel $K_{0.5}(v)$, where $\hat{A}_2 = (2, 6)$

The signal separation, outlined above, depends on correct estimation of all the maxima. However, in many cases the mixing matrix amplifies the energy of one of the signals more than the other and thus, only one maximum can be found using the above techniques. The second maximum is obscured in such cases. This is indeed the case of the probability function for the mixtures of the cameraman and tissue image depicted in Figure 25 where, the mixing matrix of 52 was used. In this case the coefficient value of 6 dominates the distribution (the maximum is actually at 5.96) but the expected maximum at 2 can not be identified. Using this provisional wrong estimate of $\hat{A}_2 = (0, 6)$ the reconstruction is now correct only for one of the signals (see Figure 29), the second reconstructed signal contains a mixture of the cameraman and tissue images. The problem happens because we estimated the second coefficient to be 0 instead of 2. None of the above mentioned techniques for finding maxima can cope with this problem. The solution can be derived from the observation that when one of the coefficients is estimated correctly, it can be used to "disable" one of the signals from the first mixture even without knowing the value of the second coefficient. Thus we begin by first finding only one maximum (using a gradient ascent or a similar optimization algorithm) - this maximum corresponds to one of the coefficients. After finding one of the coefficients (without loss of generality we assume that it is $\hat{A}_{2,2}$), we can repeat the application of the above algorithm, where instead of using the original mixtures, we now use

$$\begin{pmatrix} z_1(\xi) \\ z_2(\xi) \end{pmatrix} = \begin{pmatrix} x_1(\xi) - \frac{1}{\hat{A}_{2,2}} x_2(\xi) \\ x_2(\xi) \end{pmatrix} = \\ = \begin{pmatrix} (1 - \frac{\hat{A}_{2,1}}{\hat{A}_{2,2}}) \hat{\mathbf{s}}_1(\xi) \\ \hat{A}_{2,1} \hat{\mathbf{s}}_1(\xi) + \hat{A}_{2,2} \hat{\mathbf{s}}_2(\xi) \end{pmatrix}.$$

Using the vector $\mathbf{z}(\xi)$ as the new signals yields:

$$r_1(\xi) = \frac{\Psi_\delta \left[\hat{A}_{2,1} \hat{\mathbf{s}}_1 + \hat{A}_{2,2} \hat{\mathbf{s}}_2 \right](\xi)}{\Psi_\delta \left[(1 - \frac{\hat{A}_{2,1}}{\hat{A}_{2,2}}) \hat{\mathbf{s}}_1 \right](\xi)} \approx \\ \approx \frac{\hat{A}_{2,1} \hat{A}_{2,2} \Psi_\delta \left[\hat{\mathbf{s}}_1 \right](\xi)}{(\hat{A}_{2,2} - \hat{A}_{2,1}) \Psi_\delta \left[\hat{\mathbf{s}}_1 \right](\xi)} + \frac{\hat{A}_{2,2}^2 \Psi_\delta \left[\hat{\mathbf{s}}_2 \right](\xi)}{(\hat{A}_{2,2} - \hat{A}_{2,1}) \Psi_\delta \left[\hat{\mathbf{s}}_1 \right](\xi)}.$$

The distribution of the element $\frac{\hat{A}_{2,1} \hat{A}_{2,2} \Psi_\delta [\hat{\mathbf{s}}_1](\xi)}{(\hat{A}_{2,2} - \hat{A}_{2,1}) \Psi_\delta [\hat{\mathbf{s}}_1](\xi)}$ is approximately a Gaussian concentrated around $\frac{\hat{A}_{2,1} \hat{A}_{2,2}}{\hat{A}_{2,2} - \hat{A}_{2,1}}$. The element $\frac{\hat{A}_{2,2}^2 \Psi_\delta [\hat{\mathbf{s}}_2](\xi)}{(\hat{A}_{2,2} - \hat{A}_{2,1}) \Psi_\delta [\hat{\mathbf{s}}_1](\xi)}$ adds symmetric noise with zero mean (the signals are uncorrelated and the mean of $\Psi_\delta \left[\hat{\mathbf{s}}_2 \right](\xi)$ is 0) and thus should not alter the location of the expected maximum. Therefore, the maximum of the probability function is now located at $\frac{\hat{A}_{2,1} \hat{A}_{2,2}}{\hat{A}_{2,2} - \hat{A}_{2,1}}$.

At Figure 26 we can see that the maximum is now achieved at 4.03 and $\hat{A}_{2,1}$ can be estimated as follows:

$$\frac{\hat{A}_{2,1} \hat{A}_{2,2}}{(\hat{A}_{2,2} - \hat{A}_{2,1})} = 4.03 \quad \Rightarrow \\ \Rightarrow \hat{A}_{2,1} = \frac{4.03 \hat{A}_{2,2}}{\hat{A}_{2,2} + 4.03} = 2.4 \quad .$$

This estimation provides a much better reconstruction of the signals and, in fact, the unmixing achieves visually, close to perfect results in this case (Figure 30).

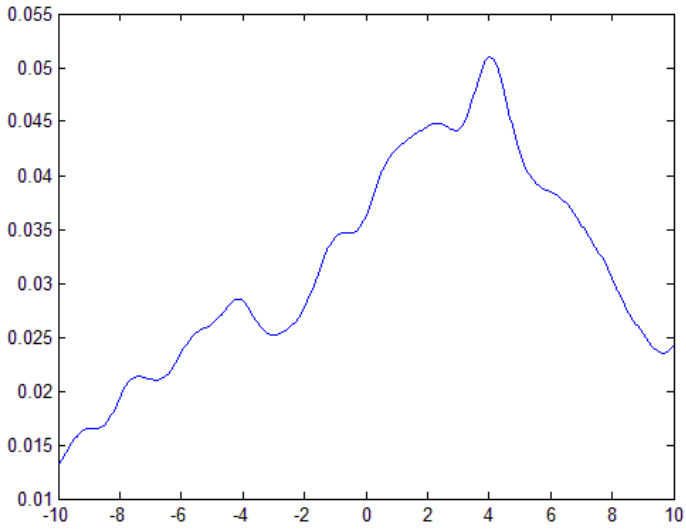

Fig. 26. p_{r_1} for mixture of two signals $z_1(\xi), z_2(\xi)$ estimated with kernel $K_{0.5}(v)$, where $\hat{A}_2 = (2, 6)$

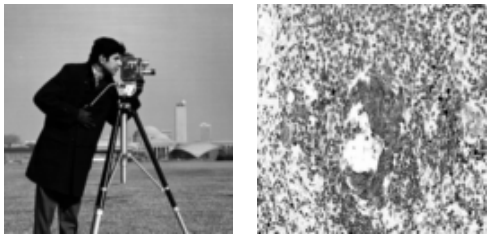

Fig. 27. Original cameraman and tissue images

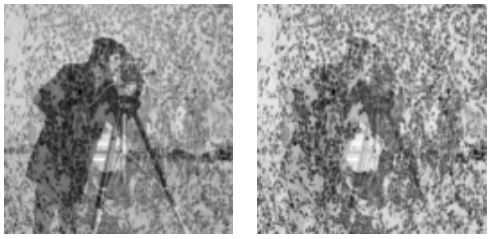

Fig. 28. Mixed images with $\hat{A}_2 = (2, 6)$

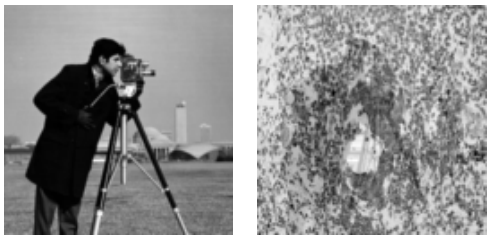

Fig. 29. Reconstructed images, based on the estimation $\hat{A}_2 = (0, 5.96)$

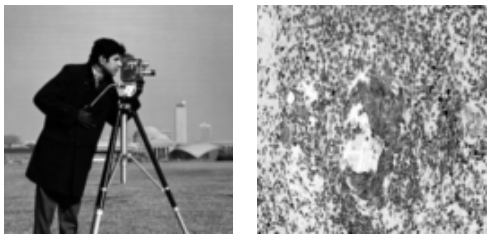

Fig. 30. Reconstructed images with maxima estimated using a two-staged approach, the resultant estimated $\hat{A}_2 = (2.4, 5.96)$ used for the reconstruction

2) Reconstruction in the general case: In the general case, where we deal with mixtures of more than two signals, similar reconstruction techniques can be applied in principle. However, the following additional difficulties usually arise:

- Ordering is now an issue - even when we know all elements in each matrix row, we do not know their order and the matrix can not be inverted. Using an arbitrary ordering is now not permitted, since the only permutations that do not affect the result, are permutations of the whole columns. Arbitrary permutations in each row lead to a wrong matrix inversion.
- Due to the interference of multiple sources, additional local maxima that do not correspond to any matrix coefficient can be found.

In this work we solve the ordering problem by estimating maximum probability of the joint distribution of vector $r = [r_1(\xi), r_2(\xi), ..., r_{M-1}(\xi)]$, where each maximum corresponds to a column of the mixing matrix $\hat{A}$. Although the approach for estimating elements of each line separately, followed by clustering of the maxima to find the columns could be used, it can not be extended and applied to the time/position varying case.

The problem of local maxima is solved by using an iterative approach. We start from the estimation of p_r which uses kernel function with large σ and perform gradient ascent with simulated annealing. At each additional iteration we perform the same optimization, but with kernel that uses smaller values of σ, where the starting point for the optimization is the maximum found at the previous iteration. This approach can be seen as finding the maximum at a coarse level, and improving the resolution of the maximum locations at next steps. The probability function that we use for the optimization is

$$p_r(\mathbf{v}) = \frac{1}{L} \sum_{\xi \in \Xi_L} K_\sigma(\mathbf{r}(\xi) - (v)), \tag{56}$$

where K_σ for the M dimensional case (where the dimension of r is $M-1$), is defined by

$$K_\sigma(v) = \frac{e^{-\frac{||v||^2}{2\sigma}}}{(2\pi)^{\frac{M-1}{2}} \sigma^{\frac{1}{2}}}. \tag{57}$$

This approach is formalized in Algorithm 2:

Algorithm 2 Calculate $\mathbf{v} = FindMax(\mathbf{r}(\xi), \Xi_L, \sigma_{init}, numIterations)$

$\mathbf{v} \leftarrow$ Random $(M-1) \times 1$ vector
$\sigma \leftarrow \sigma_{init}$
$count \leftarrow 0$
while $count < numIterations$ **do**
  $p_r \leftarrow$ approximation using K_σ
  $\mathbf{v} \leftarrow$ gradient ascent on p_r starting from $\mathbf{v}$
  $\sigma \leftarrow \frac{\sigma}{2}$
  $count \leftarrow count + 1$
end while

As in the case with two signals, this algorithm finds only one column of the matrix $\hat{A}$. In order to find all columns we perform the optimization in Algorithm 2, M times. This

iterative approach consists of multiple steps, where at iteration $m+1$, we assume that the first m columns of the matrix $\hat{A}$ were already estimated correctly, and thus, at iteration $m+1$ we can estimate the $m+1$ column of the matrix with the following steps:

1) Replace $\mathbf{x}_1(\xi)$ with a new signal $\mathbf{z}_1(\xi)$ which is a linear combination of the signals $\mathbf{x}(\xi)$, such that it is only a mixture of signals $\hat{\mathbf{s}}_{m+1,...,M}(\xi)$.
2) Sparsify the signals, and calculate $r(\xi)$, where instead of using $\mathbf{x}_1(\xi)$ in the denominator, use $\mathbf{z}_1(\xi)$ (it will be shown later that in such a case, no maximum in p_r will correspond to any of the first m columns of matrix $\hat{A}$ at step $m+1$).
3) Find maximum in p_r.
4) Reconstruct the $m+1$ column of $\hat{A}$ from the maximum of p_r.

We now explain in details how the above steps are performed. We assume that for all $m < M$ the rank of a sub-matrix $\hat{A}_{2..M,1..m}$ equals to m (for randomly generated matrices with non singular continuous pdf this assumption holds with a probability 1). If the rank of $\hat{A}_{2..M,1..m}$ is m, there must exist a subset of m linearly independent rows. For the propose of clarity we assume that the rows $2,...,m+1$ are linearly independent, although the algorithm below would work with any set of m linearly independent rows. Let Q be the invertible $m \times m$ matrix defined by:

$$Q \equiv \hat{A}_{2..m+1,1..m}, \tag{58}$$

and let G be the matrix with the remaining columns at these rows :

$$G = \hat{A}_{2..m+1,m+1..M}. \tag{59}$$

It should be noted that when the first m columns of the matrix $\hat{A}$ are known, the matrix Q is known, and the matrix G is unknown. We can define $\alpha(\xi)$ as a subset of m signals of $\mathbf{x}(\xi)$:

$$\alpha(\xi) = \mathbf{x}_{2..m+1}(\xi). \tag{60}$$

Based on this definition, it follows that $\alpha(\xi) = Q\hat{\mathbf{s}}_{1..m}(\xi) + G\hat{\mathbf{s}}_{m+1..M}(\xi)$. Let $\mathbf{w}$ be the $m \times 1$ vector defined by the equation

$$\mathbf{w} = -Q^{-T}\mathbf{1}. \tag{61}$$

Then, according to Lemma 4.2, the signal $\mathbf{w}^T\alpha(\xi) + \mathbf{x}_1(\xi)$ is a mixture only of signals $\hat{\mathbf{s}}_{m+1}(\xi), \hat{\mathbf{s}}_{m+2}(\xi), ..., \hat{\mathbf{s}}_M(\xi)$.

To illustrate the above definitions, let us look at the example for a mixture of 4 signals, where

$$\begin{pmatrix} \mathbf{x}_1(\xi) \\ \mathbf{x}_2(\xi) \\ \mathbf{x}_3(\xi) \\ \mathbf{x}_4(\xi) \end{pmatrix} = \begin{pmatrix} 1 & 1 & 1 & 1 \\ \hat{A}_{2,1} & \hat{A}_{2,2} & \hat{A}_{2,3} & \hat{A}_{2,4} \\ \hat{A}_{3,1} & \hat{A}_{3,2} & \hat{A}_{3,3} & \hat{A}_{3,4} \\ \hat{A}_{4,1} & \hat{A}_{4,2} & \hat{A}_{4,3} & \hat{A}_{4,4} \end{pmatrix} \begin{pmatrix} \hat{\mathbf{s}}_1(\xi) \\ \hat{\mathbf{s}}_2(\xi) \\ \hat{\mathbf{s}}_3(\xi) \\ \hat{\mathbf{s}}_4(\xi) \end{pmatrix},$$

then in the third iteration:

$$\begin{aligned} Q &= \begin{pmatrix} \hat{A}_{2,1} & \hat{A}_{2,2} \\ \hat{A}_{3,1} & \hat{A}_{3,2} \end{pmatrix} \\ G &= \begin{pmatrix} \hat{A}_{2,3} & \hat{A}_{2,4} \\ \hat{A}_{3,3} & \hat{A}_{3,4} \end{pmatrix} \\ \alpha(\xi) &= \begin{pmatrix} \mathbf{x}_2(\xi) \\ \mathbf{x}_3(\xi) \end{pmatrix} \end{aligned}$$

Lemma 4.2: Let Q, G, $\alpha(\xi)$, be defined as above, then $\mathbf{w}^T\alpha(\xi) + \mathbf{x}_1(\xi)$ is only a mixture of signals $\hat{\mathbf{s}}_{m+1}(\xi), \hat{\mathbf{s}}_{m+2}(\xi), ..., \hat{\mathbf{s}}_M(\xi)$.

We can now define signals transformation

$$\mathbf{z}_1(\xi) = \mathbf{w}^T\alpha_i(\xi) + \mathbf{x}_1(\xi) \tag{62}$$

$$\mathbf{z}_i(\xi) = \mathbf{x}_i(\xi) \quad i \in 2..M \tag{63}$$

The expression for $\tilde{\mathbf{r}}_i(\xi)$ when calculated for signals, $\mathbf{z}(\xi)$ now becomes

$$\begin{aligned} \tilde{\mathbf{r}}_i(\xi) &= \frac{\Psi_\delta\left[\mathbf{z}_i\right](\xi)}{\Psi_\delta\left[\mathbf{z}_1\right](\xi)} = \\ &= \frac{\Psi_\delta\left[\mathbf{x}_i\right](\xi)}{\Psi_\delta\left[\mathbf{w}^T\alpha(\xi) + \mathbf{x}_1(\xi)\right]} = \\ &= \frac{\Psi_\delta\left[\hat{A}_i\hat{\mathbf{s}}\right](\xi)}{\Psi_\delta\left[\mathbf{w}^T(Q\hat{\mathbf{s}}_{1..m}(\xi) + G\hat{\mathbf{s}}_{m+1..M}(\xi)) + \mathbf{x}_1(\xi)\right]} \quad . \end{aligned}$$

Note, that now no maxima of $p_{\tilde{r}}$ will correspond to the first m signals (and thus to the first m columns of $\hat{A}$), since the signals $\hat{\mathbf{s}}_{1..m}$ no longer appear in the mixture $\mathbf{z}_1(\xi)$. When Algorithm 2 is applied to $\tilde{\mathbf{r}}(\xi)$, without loss of generality, we can assume that the estimated location of maximum $\mathbf{v}$ corresponds to the $(m+1)$th column of matrix[6] $\hat{A}$ (the order of the columns does not matter, since the reconstruction is correct only up to permutations).

Let us define $\mathbf{c} = G_{1..m,1} = \hat{A}_{2..m+1,m+1}$. At points that contribute to the maximum v, only the $m+1$ source has high values and all other sources have almost zero value. Therefore, similar to the two-dimensional case :

$$\begin{aligned} v_i &= \frac{\Psi_\delta\left[\hat{A}_{i+1}\hat{\mathbf{s}}\right](\xi)}{\Psi_\delta\left[\mathbf{w}^T(Q\hat{\mathbf{s}}_{1..m}(\xi) + G\hat{\mathbf{s}}_{m+1..M}(\xi)) + \mathbf{x}_1(\xi)\right]} \approx \\ &\approx \frac{\Psi_\delta\left[\hat{A}_{i+1,m+1}\hat{\mathbf{s}}_{m+1}\right](\xi)}{\Psi_\delta\left[\mathbf{w}^T\mathbf{c}\hat{\mathbf{s}}_{m+1}(\xi) + \hat{\mathbf{s}}_{m+1}(\xi)\right]} \approx \\ &\approx \frac{\hat{A}_{i+1,m+1}}{\mathbf{w}^T\mathbf{c}+1}. \end{aligned}$$

Recalling that $\hat{A}_{i+1,m+1} = \mathbf{c}_i; \quad \forall i \in 1..m$, values of the vector $\mathbf{c}$ can now be found by solving a set of linear equations:

$$\mathbf{v}_i = \frac{\mathbf{c}_i}{\mathbf{w}^T\mathbf{c}+1}.$$

Let us denote by W a matrix all rows of which are equal to $\mathbf{w}^T$, then

$$\begin{aligned} \mathrm{diag}(\mathbf{v}_{1..m})(W\mathbf{c}+\mathbf{1}) &= \mathbf{c} \\ \mathbf{c} &= (I - \mathrm{diag}(\mathbf{v}_{1..m})W)^{-1}\mathbf{v}_{1..m} \quad . \end{aligned}$$

[6]Note that the dimensions of v are $M-1 \times 1$

The $(m+1)$th column of matrix $\hat{A}$ can now be easily reconstructed as

$$\hat{A}_{2:M,m+1} = \mathbf{v}(\mathbf{w}^T\mathbf{c}+1) \quad . \tag{64}$$

Based on the above, the high level algorithm for the reconstruction of matrix $\hat{A}$ is outlined by Algorithm 3.

Algorithm 3 Calculate $\hat{A} = ReconstructMixingMatrix(\mathbf{x}(\xi), \sigma_{init}, numIterations)$

$\hat{A} \leftarrow M \times M$ matrix of ones
$[\mathbf{r}(\xi), \Xi_L] \leftarrow SparsifyAndFindR(\mathbf{x}(\xi))$
$\hat{A}_{2..m,1} \leftarrow FindMax(\mathbf{r}(\xi), \Xi_L, \sigma_{init}, numIterations)$
$m \leftarrow 1$
$\mathbf{z}(\xi) \leftarrow \mathbf{x}(\xi)$
while $m < M$ **do**
  $Q \leftarrow \hat{A}_{2..m+1,1..m}$
  $\alpha(\xi) \leftarrow \mathbf{x}_{2..m+1}(\xi)$
  $\mathbf{w} \leftarrow -Q^{-T}\mathbf{1}$
  $\mathbf{z}_1(\xi) \leftarrow \mathbf{w}^T\alpha_i(\xi) + \mathbf{x}_1(\xi)$
  $[\tilde{\mathbf{r}}(\xi), \tilde{\Xi_L}] \leftarrow SparsifyAndFindR(\mathbf{z}(\xi))$
  $\mathbf{v} \leftarrow FindMax(\tilde{r}(\xi), \tilde{\Xi_L}, \sigma_{init}, numIterations)$
  $\mathbf{c} \leftarrow (I - \mathrm{diag}(\mathbf{v}_{1..m})W)^{-1}\mathbf{v}_{1..m}$
  $\hat{A}_{2..M,m+1} \leftarrow \mathbf{v}(\mathbf{w}^T\mathbf{c}+1)$
  $m \leftarrow m+1$
end while

3) Results: Algorithm 3 performs well on most of mixing matrices. Figure 33 illustrates an example of reconstruction for 4 signals. We used the following, randomly generated mixing matrix:

$$\hat{A} = \begin{pmatrix} 1 & 1 & 1 & 1 \\ 0.9262 & -1.2253 & 4.4888 & 0.1447 \\ 1.7310 & -0.8314 & -2.2299 & 1.3705 \\ 2.0753 & 3.6444 & -1.0580 & -3.2559 \end{pmatrix}. \tag{65}$$

The mixing matrix, estimated by the algorithm, is

$$\hat{A} \approx \begin{pmatrix} 1 & 1 & 1 & 1 \\ 0.4696 & 4.7900 & -1.1559 & 0.1560 \\ 2.1144 & -2.2772 & -0.6993 & 1.3828 \\ 1.5624 & -1.3248 & 3.2993 & -3.3289 \end{pmatrix}. \tag{66}$$

The estimation of the matrix is close to the original one (up to permutation of columns), and provides visually good reconstruction of the original images (Figure 33).

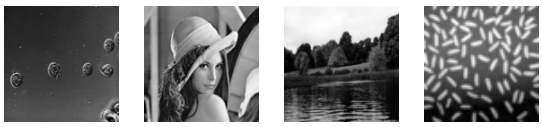

Fig. 31. Original images ("sources")

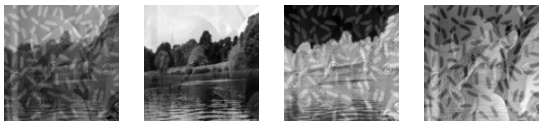

Fig. 32. Four mixtures of images shown in Figure 31

Fig. 33. Reconstructed (separated) images

The reconstruction was performed using gradient descent with the use of multiple restarts and simulated annealing.

V. SSCA of Time/Position Varying Mixtures

We make the following assumptions about the mixing model:

- The mixed signals are independent.
- The mixing model is the time/position varying model, as defined by (2), wherein the parameter vector θ_{mix} from which the mixtures were generated is unknown.
- The parametric model of the mixing matrix $A(\xi,\theta)$ is known and the mixing matrix vary slowly with the time/position parameter - for small values of ϵ if $||\xi_1 - \xi_2||_2 < \epsilon$, then $A(\xi_1,\theta) \approx A(\xi_2,\theta) \quad \forall\theta$. From this assumption it follows that $\Psi_\delta[\mathbf{x}](\xi) \approx A(\xi,\theta)\Psi_\delta[\mathbf{s}](\xi)$.
- All elements in the first row of the mixing matrix $A(\xi,\theta)$ are non-zero for most of the values of ξ.

As in the case of time/position invariant mixing systems, our goal in this section is to reconstruct the signals up to a distortion, i.e, instead of reconstructing $\mathbf{s}_i(\xi)$; $\forall i$, we focus on the reconstruction of the distorted signals $\hat{\mathbf{s}}_i(\xi) \equiv A_{1,i}(\xi,\theta_{mix})\mathbf{s}_i(\xi)$. Thus, from here on, we can assume that our mixing model is:

$$\mathbf{x}(\xi) = \hat{A}(\theta_{mix},\xi)\hat{\mathbf{s}}(\xi),$$

where

$$\hat{A}_{i,j}(\theta_{mix},\xi) = \frac{A_{i,j}(\theta_{mix},\xi)}{A_{1,j}(\theta_{mix},\xi)} \quad \forall j \in 1..M.$$

We first present the algorithm for reconstruction in the case of mixtures of two signals, and later extend the algorithm to deal with mixtures of more signals.

A. SSCA of Time/Position Varying Mixtures of Two Signals

In the case of two signals, our general mixing model takes the form of

$$\begin{pmatrix} \mathbf{x}_1(\xi) \\ \mathbf{x}_2(\xi) \end{pmatrix} = \begin{pmatrix} 1 & 1 \\ \hat{A}_{2,1}(\xi,\theta_{mix}) & \hat{A}_{2,2}(\xi,\theta_{mix}) \end{pmatrix} \begin{pmatrix} \hat{\mathbf{s}}_1(\xi) \\ \hat{\mathbf{s}}_2(\xi) \end{pmatrix}. \tag{67}$$

Let Ξ_L be defined in the same way as in the time/position invariant case. We observe that for all $\xi \in \Xi_L$:

$$\begin{aligned} \mathbf{r}(\xi) &\equiv \frac{\Psi_\delta[\mathbf{x}_2(\xi)]}{\Psi_\delta[\mathbf{x}_1(\xi)]} = \\ &= \frac{\Psi_\delta\left[\hat{A}_{2,1}(\xi,\theta_{mix})\hat{\mathbf{s}}_1(\xi) + \hat{A}_{2,2}(\xi,\theta_{mix})\hat{\mathbf{s}}_2(\xi)\right]}{\Psi_\delta[\hat{\mathbf{s}}_1(\xi) + \hat{\mathbf{s}}_2(\xi)]} \approx \\ &\approx \frac{\hat{A}_{2,1}(\xi,\theta_{mix})\Psi_\delta[\hat{\mathbf{s}}_1(\xi)] + \hat{A}_{2,2}(\xi,\theta_{mix})\Psi_\delta[\hat{\mathbf{s}}_2(\xi)]}{\Psi_\delta[\hat{\mathbf{s}}_1(\xi)] + \Psi_\delta[\hat{\mathbf{s}}_2(\xi)]} \end{aligned} \tag{68}$$

By applying Lemma 4.1, (68) yields

$$\mathbf{r}(\xi) \approx \hat{A}_{2,i}(\xi,\theta_{mix}); \quad \exists i \in \{1,2\} \quad . \tag{69}$$

Let us define two probabilities μ_i as

$$\mu_i \equiv p(\mathbf{r}(\xi) = \hat{A}_{2,i}(\xi,\theta_{mix})); \quad i \in \{1,2\}. \tag{70}$$

If we assume that the matrix $\hat{A}(\xi,\theta_{mix})$ is invertible for all ξ, then it is clear that $\mu_1 + \mu_2 = 1$ ($\mathbf{r}(\xi)$ is equal either to the first or to the second element, but never to both at the same time). Let us approximate the conditional probability density functions

$$f_i(\theta) \equiv p(\mathbf{r}(\xi) = \hat{A}_{2,i}(\xi,\theta)|\theta) \quad i \in \{1,2\} \tag{71}$$

as

$$f_i(\theta) = \frac{1}{L}\sum_{\xi\in\Xi_L} K_\sigma(\hat{A}_{2,i}(\xi,\theta) - r(\xi)) \quad i \in \{1,2\}, \tag{72}$$

where K_σ is a Gaussian defined by (53). Let $F_\sigma(\theta)$ be the sum of the above conditional probabilities:

$$F_\sigma(\theta) = f_1(\theta) + f_2(\theta), \tag{73}$$

then according to Lemma 5.1, for small enough σ, the global maximum of $F_\sigma(\theta)$ is located near θ_{mix}, under reasonable assumptions.

Lemma 5.1: Let $F_\sigma(\theta)$ be defined by (73). If the following conditions hold

1) For a given sparsification and a given parametric family, (69) holds for all $\xi \in \Xi_L$,
2) $\hat{A}(\xi,\theta)$ is invertible for all $\xi \in \Xi_L$,
3) $p\left(\hat{A}_{2,i}(\xi,\theta_{\mathbf{1}}) = \hat{A}_{2,j}(\xi,\theta_{\mathbf{2}})\right) = 0$, for $i \neq j$, $\forall\theta_1,\theta_2$,

then $\lim_{\sigma\to 0}\operatorname{argmax}\{F_\sigma(\theta)\} = \theta_{\mathbf{mix}}$.

Proof: Let μ_1 and μ_2 be defined by (70), then :

$$\begin{aligned}
F_\sigma(\theta) &= f_1(\theta)+f_2(\theta)= \\
&= \sum_{\xi\in\Xi_L} K_\sigma(\hat{A}_{2,1}(\xi,\theta)-r(\xi))+ \\
&\quad +\sum_{\xi\in\Xi_L} K_\sigma(\hat{A}_{2,2}(\xi,\theta)-r(\xi))= \\
&= \mu_1\sum_{\xi\in\Xi_L} K_\sigma(\hat{A}_{2,1}(\xi,\theta)-\hat{A}_{2,1}(\xi,\theta_{\mathbf{mix}})) \\
&\quad +\mu_2\sum_{\xi\in\Xi_L} K_\sigma(\hat{A}_{2,1}(\xi,\theta)-\hat{A}_{2,2}(\xi,\theta_{\mathbf{mix}}))+ \\
&\quad +\mu_1\sum_{\xi\in\Xi_L} K_\sigma(\hat{A}_{2,2}(\xi,\theta)-\hat{A}_{2,1}(\xi,\theta_{\mathbf{mix}})) \\
&\quad +\mu_2\sum_{\xi\in\Xi_L} K_\sigma(\hat{A}_{2,2}(\xi,\theta)-\hat{A}_{2,2}(\xi,\theta_{\mathbf{mix}}))= \\
&= \mu_1\sum_{\xi\in\Xi_L} K_\sigma(\hat{A}_{2,1}(\xi,\theta)-\hat{A}_{2,1}(\xi,\theta_{\mathbf{mix}}))+ \\
&\quad +K_\sigma(\hat{A}_{2,2}(\xi,\theta)-\hat{A}_{2,1}(\xi,\theta_{\mathbf{mix}}))+ \\
&\quad +\mu_2\sum_{\xi\in\Xi_L} K_\sigma(\hat{A}_{2,2}(\xi,\theta)-\hat{A}_{2,2}(\xi,\theta_{\mathbf{mix}}))+ \\
&\quad +K_\sigma(\hat{A}_{2,1}(\xi,\theta)-\hat{A}_{2,2}(\xi,\theta_{\mathbf{mix}})).
\end{aligned}$$

According to the last condition,

$$\begin{aligned}
p\left(\hat{A}_{2,2}(\xi,\theta)=\hat{A}_{2,1}(\xi,\theta_{\mathbf{mix}})\right) &= 0 \\
p\left(\hat{A}_{2,1}(\xi,\theta)=\hat{A}_{2,2}(\xi,\theta_{\mathbf{mix}})\right) &= 0.
\end{aligned}$$

In addition, we note that :

$$\lim_{\sigma\to 0} K_\sigma(v)=0 \quad \forall v\neq 0.$$

Thus,

$$\begin{aligned}
&\lim_{\sigma\to 0}\mu_1\sum_{\xi\in\Xi_L} K_\sigma(\hat{A}_{2,2}(\xi,\theta)-\hat{A}_{2,1}(\xi,\theta_{\mathbf{mix}}))=0, \\
&\lim_{\sigma\to 0}\mu_2\sum_{\xi\in\Xi_L} K_\sigma(\hat{A}_{2,1}(\xi,\theta)-\hat{A}_{2,2}(\xi,\theta_{\mathbf{mix}}))=0,
\end{aligned}$$

and

$$\lim_{\sigma\to 0}\operatorname{argmax}\{F_\sigma(\theta)\}=\theta_{\mathbf{mix}}.$$

■

Note that in examples adopted from real life, the assumptions of Lemma. 5.1 usually hold only approximately and, of course we use a finite non-zero (but small) value of σ. Therefore, we expect that the maximum may be achieved not at $\theta_{\mathbf{mix}}$ but close to it. For 2 signals the reconstruction is performed using θ^*, which maximizes the $F_\sigma(\theta)$ expression in (73). The optimization is performed for a few iterations, where for each iteration the value of σ is decreased.

As in the time/position invariant case, it is possible that only one of the columns is dominant; for example when $\mu_2 = 0$. It happens if $f_1(\theta^*)$ is significantly larger than $f_2(\theta^*)$. In this case θ^* might approximate one of columns correctly, but produce incorrect approximation for the second column. We can perform the same manipulation as in the time/position invariant case. Define

$$\begin{aligned}
\mathbf{z}_1(\xi) &= \mathbf{x}_1(\xi)-\frac{1}{\hat{A}_{2,1}(\xi,\theta^*)}\mathbf{x}_2(\xi), \\
\mathbf{z}_2(\xi) &= \mathbf{x}_2(\xi), \\
\tilde{\mathbf{r}}(\xi) &= \frac{\Psi_\delta\left[\mathbf{z}_2(\xi)\right]}{\Psi_\delta\left[\mathbf{z}_1(\xi)\right]}.
\end{aligned}$$

The energy function that we maximize in the second step is

$$\tilde{F}_\sigma(\theta)=\sum_{\xi\in\Xi_L} K_\sigma\left(\tilde{\mathbf{r}}(\xi)-\frac{\hat{A}_{2,2}(\xi,\theta)}{\hat{A}_{2,1}(\xi,\theta^*)}\right). \tag{74}$$

Based on our experiments, this function is far from being smooth for most of the parametric families and the optimization of this function does not provide good results. This happens due to the fact that the points that pass the threshold during the sparsification process are mostly points for which $\hat{A}_{2,1}(\xi,\theta^*)$ has small values (and thus $\frac{1}{\hat{A}_{2,1}(\xi,\theta^*)}$ is large). In this case, small deviations in $\hat{A}_{2,2}(\xi,\theta)$ cause a significant change in the value of $\frac{\hat{A}_{2,2}(\xi,\theta)}{\hat{A}_{2,1}(\xi,\theta)}$ and the optimal θ is estimated incorrectly. Thus the SSCA solution for parametric families with μ_1 or μ_2 that are close to 0, remains an open problem.

B. SSCA of Time/Position Varying Mixtures of Multiple Signals

The approach applied so far for two signals, can easily be extended to the general case. We define $\mathbf{r}(\xi)$ by

$$\mathbf{r}_i(\xi)\equiv\frac{\Psi_\delta\left[\mathbf{x}_{i+1}(\xi)\right]}{\Psi_\delta\left[\mathbf{x}_1(\xi)\right]}. \tag{75}$$

Let Ξ_L be a set of L samples locations, such that $\Psi_\delta\left[\mathbf{x}_1(\xi)\right]$ $\forall\xi\in\Xi_L$. The extension of the energy function from the case of 2 signals to the case of M signals produces the energy function:

$$F_\sigma(\theta)=\sum_{i=1}^{M}\sum_{\xi\in\Xi_L} K_\sigma(\mathbf{r}(\xi)-\hat{A}_{2..M-1,i}(\xi,\theta)). \tag{76}$$

The full algorithm is presented at Algorithm 4. The mathematical justification for this algorithm is the same as in the case with two signals. For $\sigma\to 0$, in an ideal case, we expect that the optimization would achieve its optimum at $\theta^*=\theta_{\mathbf{mix}}$.

Algorithm 4 Calculate $\theta^* = FindMax(\mathbf{r}(\xi),\Xi_L,\sigma_{init},numIterations)$

$\theta^* \leftarrow$ Random vector
$\sigma \leftarrow \sigma_{init}$
$count \leftarrow 0$
while $count < numIterations$ **do**
  $p_r \leftarrow$ approximation using K_σ
  $\theta^* \leftarrow$ gradient ascent on $F_\sigma(\theta)$ starting from θ^*
  $\sigma \leftarrow \frac{\sigma}{2}$
  $count \leftarrow count+1$
end while

1) Results: Based on our experiments, we may conclude that the algorithm performs well on some of the parametric families, but fails to provide a correct reconstruction for others. One of the main problems of the algorithm is that it tries to estimate the parameters using only a small set of samples (the samples that passed a threshold). At Figure 36 we can see that the algorithm provides good reconstruction results for the parametric family

$$\hat{A}(c,r,\theta_1,\theta_2)=\begin{pmatrix}1 & 1 & 1\\ \theta_1 r^2 & \theta_2 c & 1\\ \theta_2 c^2 & 1 & \theta_1 r\end{pmatrix}, \tag{77}$$

where r and c are row and column indices normalized by a constant. In this example

$$\theta_{\mathbf{mix}}=\begin{pmatrix}6.1770\\ -7.0007\end{pmatrix} \tag{78}$$

Fig. 34. Original "source" images

Fig. 35. The three mixtures

Fig. 36. Reconstructed images

VI. Discussion And Recommendations

In this study we proposed two methods for unmixing signals/images mixtures that vary with time/position parameter. It seems that ML reconstruction method performs better than the SSCA approach on most of the parametric families. On the other hand, the ML approach can suffer from being highly nonlinear and the energy function that is optimized under the ML approach may consequently have numerous local maxima. It is very useful to test whether a specific maximum of the ML energy function is local or global maximum. If the maximum is local, then the temperature of the simulated annealing may increase in order to avoid the maximum. We believe that although SSCA method by itself does not provide good reconstruction, methods based on SSCA can be used to test whether a specific θ^* is indeed equal to $\theta_{\mathbf{mix}}$. This test can be developed based on the observation that given the approximations of signals $\mathbf{s}(\xi)$. We can observe at the matrix $R(\xi)$ defined as:

$$R_{i,j}(\xi)=\frac{\Psi_\delta\left[\mathbf{x}_j(\xi)\right]}{B_i(\xi,\theta^*)\Psi_\delta\left[\mathbf{x}(\xi)\right]}, \tag{79}$$

where $B(\xi,\theta^*)=A(\xi,\theta^*)^{-1}$. The expression in (79) is very similar to the expression used in calculation of $\mathbf{r}(\xi)$ by the means of the SSCA approach. The difference is that in the denominator we now use the estimated reconstruction of the signals instead of $\mathbf{x}_1(\xi)$. If the assumptions of the SSCA approach hold, then we can expect that $R(\xi)\approx A(\xi,\theta_{mix})$. Thus, measuring the similarity between such $R(\xi)$ and $A(\xi,\theta^*)$ can provide a measure of the optimality of θ^*. The proposed measure of optimality can not be used as an energy function by itself due to the fact that the denominator can not be differentiated by θ since it involves hard threshold.

We conclude that additional research should be devoted to the application of the SSCA method in separation of time/position varying mixtures. Issues that should be addressed in further research include:

- Multiple sparsification transformations, incorporated into the SSCA algorithm. Various specific sparsification transformations may be sensitive to different features of images more than other sparsification and when combined can provide a larger and better set of data in that it better satisfies the sparsification properties outlined in Section IV-A.
- For some parametric families, additional preprocessing of the given signals could be applied before performing the sparsification. Such preprocessing should be developed specifically for each family. For example for time/position invariant case a whitening preprocessing improves the results dramatically. For other parametric families different types of preprocessing could be developed.
- Calculation of additional $r(t)$ signals could be used, where in denominator instead of $\Psi_\delta\left[\mathbf{x}_1(\xi)\right]$ other signals ($\Psi_\delta\left[\mathbf{x}_i(\xi)\right]$, $i\neq 1$) could be used. This can provide additional data that can be used for the optimization.
- Combined ML/SSCA algorithm could be used, where ML algorithm is applied on the sparsified images (with an appropriate a priori sparse pdf function assumption).

Another important aspect of the problem that remains open is for which parametric families the reconstruction can be performed using the ML or SSCA approach and for which parametric families one should expect that the unmixing will fail. The main problems that can cause wrong unmixing are:

- Estimated statistical properties like mean/variance that are biased primarily on a small number of samples are unreliable.
- Perfect reconstruction parameters are not achieved in global maximum of the energy function. This can happen when the reconstructed signals do not satisfy some of the algorithm assumptions.
- Multiple global maxima exist for the energy function that is used for the reconstruction. This usually happens when the problem is ill posed.

Based on experiments conducted with various examples, and on basic considerations of the fundamental problem, we believe that in general, no answer exists to this question, since the unmixing is very dependent on the probability distribution and sparsification of the unknown mixed signals. Even for TPIBSS most of the existing techniques fail in some cases of

signal mixtures. On the other hand, after applying specific unmixing algorithm, it is possible to develop various tests that can verify whether the reconstructed signals and mixing matrix satisfy all the basic assumptions on which the algorithm is based. As a simple example of such a test, correlation measure of the reconstructed signals can be used. If there is a high correlation between the reconstructed signals, then with a high probability we may say that the reconstruction technique has failed and another algorithm should be applied for the given mixtures. As for the SSCA reconstruction as an example of posteriori test, we can check whether for each ξ after sparsification only one reconstructed signal has large value. If this is not the case, then with high probability we say that the SSCA-based reconstruction failed. Using such test techniques multiple unmixing algorithms could be combined, where each algorithm is applied separately and then the reconstructed signals are chosen from the algorithm that provides best posteriori test results.

In this research we used the ML and SSCA techniques for signals unmixing. Although these are two of the most popular algorithms for BSS, other unmixing algorithms exist. Additional algorithms based on block decorrelation or similar techniques [22], [23], could be used for cases with known parametric family of mixtures. Somewhat better reconstruction for large number of parametric families may be achieved using a combination of multiple unmixing algorithms, but this was not the purpose of the research outlined in this thesis. Instead, we were concerned with the more fundamental issue of whether either one of the ML and SSCA technique can be extended and applied to TPVBSS, and how the performance of these two techniques compares. This goal was accomplished with a positive result.

REFERENCES

[1] E. Be'ery and Arie Yeredor. Blind separation of superimposed shifted images using parameterized joint diagonalization. *IEEE Transactions on Image Processing*, 17(3):340–353, 2008.

[2] Anthony J. Bell and Terrence J. Sejnowski. An information-maximization approach to blind separation and blind deconvolution. *Neural Computation*, 7:1129–1159, 1995.

[3] Dimitri P. Bertsekas and Dimitri P. Bertsekas. *Nonlinear Programming*. Athena Scientific, Nashua, 2nd edition, 1999.

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

[5] A.M. Bronstein, M.M Bronstein, M. Zibulevsky, and Y.Y Zeevi. Blind deconvolution of images using optimal sparse representations. *IEEE Transactions on Image Processing*, 14:726–736, 2005.

[6] J. F. Cardoso. Infomax and maximum likelihood for blind source separation. *IEEE Signal Processing Letters*, 4(4):112–114, April 1997.

[7] J.-F. Cardoso. Blind signal separation : Statistical principles. *Proceedings of the IEEE*, 9:2009–2025, 1998.

[8] Thomas M. Cover and Joy A. Thomas. *Elements of information theory*. Wiley-Interscience, New York, 1991.

[9] Aapo Hyvärinen. Fast and robust fixed-point algorithms for independent compoent analysis. *IEEE Transactions on Neural Networks*, 10:624–634, 1999.

[10] Aapo Hyvärinen. The fixed-point algorithm and maximum likelihood estimation for independent component analysis. *Neural Processing Letters*, 10:1–5, 1999. 10.1023/A:1018647011077.

[11] Aapo Hyvärinen. Survey on independent component analysis. *Neural Computing Surveys*, 2:94–128, 1999.

[12] Aapo Hyvärinen, Juha Karhunen, and Erkki Oja. *Independent Component Analysis*. Wiley-Interscience Publication, New York, 2001.

[13] Aapo Hyvärinen and Erkki Oja. Independent component analysis: algorithms and applications. *Neural Networks*, 13(4-5):411–430, 2000.

[14] Shun ichi Amari, Scott C. Douglas, Andrzej Cichocki, and Howard H. Yang. Multichannel blind deconvolution and equalization using the natural gradient. In *In The First Signal Processing Workshop on Signal Processing Advances in Wireless Communications*, pages 101–104, 1997.

[15] Ran Kaftory. *Blind Separation of Time/Position Varying Mixtures*. PhD thesis, Techion, Haifa, Israel, 2009.

[16] Ran Kaftory, Yoav Y. Schechner, and Yehoshua Y. Zeevi. Variational distance-dependent image restoration. In *CVPR*. IEEE Computer Society, 2007.

[17] Ran Kaftory and Yehoshua Y. Zeevi. Probabilistic geometric approach to blind separation of time-varying mixtures. In Mike E. Davies, Christopher J. James, Samer A. Abdallah, and Mark D. Plumbley, editors, *ICA*, volume 4666 of *Lecture Notes in Computer Science*, pages 373–380. Springer, 2007.

[18] Ran Kaftory and Yehoshua Y. Zeevi. Blind separation of images obtained by spatially-varying mixing system. In *ICIP*, pages 2604–2607, 2008.

[19] Ran Kaftory and Yehoshua Y. Zeevi. Blind separation of position varying mixed images. In *ICIP*, pages 3913–3916. IEEE, 2009.

[20] S. Kirkpatrick, C. D. Gelatt, Jr, and M. P. Vecchi. Optimization by simulated annealing. *Science*, 220:671–680, 1983.

[21] Dinh Tuan Pham. Separation of a mixture of independent sources through a maximum likelihood approach. In *In Proc. EUSIPCO*, pages 771–774, 1992.

[22] Bernard Sarel and Michal Irani. Separating transparent layers through layer information exchange. In *ECCV 2004*, pages 328–341. Springer-Verlag, 2004.

[23] Petr Tichavský, Arie Yeredor, and Zbynek Koldovský. A fast asymptotically efficient algorithm for blind separation of a linear mixture of block-wise stationary autoregressive processes. In *ICASSP*, pages 3133–3136. IEEE, 2009.

[24] Tzahi Weisman and Arie Yeredor. Separation of periodically time-varying mixtures using second-order statistics. In Justinian P. Rosca, Deniz Erdogmus, José Carlos Príncipe, and Simon Haykin, editors, *ICA*, volume 3889 of *Lecture Notes in Computer Science*, pages 278–285. Springer, 2006.

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