ELSEVIER

Signal Processing 81 (2001) 2353–2362

SIGNAL PROCESSING

www.elsevier.com/locate/sigpro

Underdetermined blind source separation using sparse representations

Pau Bofill[a, *, 1], Michael Zibulevsky[b, 2, 3]

[a] *Dept. d'Arquitectura de Computadors, UPC, Jordi Girona, 1-3. Mòdul D6 Campus Nord, 08034 Barcelona, Spain*
[b] *Department of Computer Science, University of New Mexico, Albuquerque, NM 87131, USA*

Received 24 August 2000; received in revised form 20 June 2001

Abstract

The scope of this work is the separation of N sources from M linear mixtures when the underlying system is *underdetermined*, that is, when $M < N$. If the input distribution is *sparse* the mixing matrix can be estimated either by external optimization or by *clustering* and, given the mixing matrix, a minimal l_1 norm representation of the sources can be obtained by solving a low-dimensional linear programming problem for each of the data points. Yet, when the signals per se do not satisfy this assumption, sparsity can still be achieved by realizing the separation in a sparser transformed domain. The approach is illustrated here for $M = 2$. In this case we estimate both the number of sources and the mixing matrix by the maxima of a potential function along the circle of unit length, and we obtain the minimal l_1 norm representation of each data point by a linear combination of the pair of basis vectors that *enclose* it. Several experiments with music and speech signals show that their time-domain representation is not sparse enough. Yet, excellent results were obtained using their short-time Fourier transform, including the separation of up to six sources from two mixtures.

Keywords: Blind source separation; Underdetermined source separation; Sparse signal representation; Potential-function clustering; l_1 norm decomposition

1. Blind source separation with more sources than mixtures

Let $\mathbf{x}^t$ be an M-dimensional column vector corresponding to the output of M sensors at a given discrete time instant t, and let $\mathbf{X}$ be an $M \times T$ matrix corresponding to the sensor data at all times $t = 1, \ldots, T$ (i.e., row i of $\mathbf{X}$, denoted by $\mathbf{X}_{i.}$ corresponds to the ith mixture signal). Let $\mathbf{S}$ be the $N \times T$ matrix of underlying source signals and let $\mathbf{A}$ be the $M \times N$ mixing matrix. The problem of *blind source separation* [7], in the noiseless case, consists

* Corresponding author.
E-mail addresses: pau@ac.upc.es (P. Bofill), mzib@ee.technion.ac.il (M. Zibulevsky).

[1] With support from ajuts BE98/99, DGR-Generalitat de Catalunya.

[2] Supported by NSF CAREER award 97-02-311, the National Foundation for Functional Brain Imaging, an equipment grant from Intel corporation, the Albuquerque High Performance Computing Center, a gift from George Cowan, and a gift from the NEC Research Institute.

[3] Currently at the Faculty of Electrical Engineering, Technion-Israel Institute of Technology, Haifa 32000, Israel.

0165-1684/01/$ - see front matter
PII: S0165-1684(01)00120-7

of finding the solution to the following system of equations:

$$\mathbf{X} = \mathbf{AS}, \tag{1}$$

when $\mathbf{A}$ and $\mathbf{S}$ are *unknown* ($\mathbf{A}$ will be assumed to be of full rank). For our purposes, a useful formulation of this system is obtained by decomposing $\mathbf{A}$ into its columns $\mathbf{a}^j$ and expanding for every data point

$$\mathbf{x}^t = \sum_{j=1}^{N} \mathbf{a}^j s_j^t \quad \text{for } t = 1, \ldots, T. \tag{2}$$

Then, in the M-dimensional mixture space, the $\mathbf{a}^j$'s are the basis vectors defining the spatial signature of the sources and the s_j^t's are the corresponding *components* of the data points. Since results are not affected by reciprocal rescaling of the $\mathbf{a}^j$'s and the s_j^t's, without loss of generality the $\mathbf{a}^j$'s will be hereafter assumed to be normalized to unit length.

For $M = N$, several approaches to *independent component analysis* have been used in the literature (see for instance [6] for a recent survey) to numerically solve Eq. (1) while assuming only statistical independence of the source components s_j^t. Of particular interest to the work presented here is the so-called *sparse* case, in which only a small number of the s_j^t's differ significantly from zero. Sparsity is often modeled by a Laplacian distribution [11].

One of the most popular ICA approaches is the InfoMax algorithm [1] that maximizes the information of the recovered sources. When specialized for the Laplacian distribution, it leads to the following objective function:

$$\min_{\mathbf{W}} \; -T \log |\det \mathbf{W}| + \sum_{jt} |\mathbf{WX}|_{jt} \tag{3}$$

with $\mathbf{W}$ being the estimate of $\mathbf{A}^{-1}$, $(\mathbf{WX})_{jt} = s_j^t$ the estimates of the source components, and $|\cdot|$ denoting the absolute value.

The drawback of the above formulation is that it assumes the existence of the inverse matrix $\mathbf{W}$. Therefore, it is unsuitable for the *underdetermined* case $M < N$. The alternative is to formulate the search in mixing space rather than separation space. Generalizing Eq. (1) to the case with additive Gaussian noise $\mathbf{X} = \mathbf{AS} + \mathbf{V}$, and assuming that $\mathbf{A}$ is uniformly distributed, a maximum a posteriori log-probability analysis leads to the following objective function [11,14]:

$$\min_{\mathbf{A},\mathbf{S}} \frac{1}{2\sigma^2} ||\mathbf{AS} - \mathbf{X}||^2 + \sum_{jt} |s_j^t| \tag{4}$$

with σ^2 the variance of the noise $\mathbf{V}$. The first term is the sum squared reconstruction error (the log-likelihood of the Gaussian noise), and the second term is the penalty for non-sparsity (assuming independent Laplacian sources).

An overview of different approaches to underdetermined BSS may be found in [4].

2. Estimating the mixing matrix and the sources separately

As opposed to the case of a square mixing matrix, where finding $\mathbf{W}$ amounts to solving the problem $\mathbf{S} = \mathbf{WX}$, in the underdetermined case we are faced with *two* interrelated problems: estimating the mixing matrix $\mathbf{A}$ *and* estimating the sources $\mathbf{S}$. Trying to solve both of them at the same time as in Eq. (4) is a difficult multivariate optimization problem.

Yet if we assume that the matrix $\mathbf{A}$ is given, the problem of inferring the sources can be formulated *independently* for each data point x^t, leading to T tractable small problems

$$\min_{\mathbf{s}^t} \frac{1}{2\sigma^2} ||\mathbf{A}\mathbf{s}^t - \mathbf{x}^t||^2 + \sum_{j} |s_j^t| \quad \text{for } t = 1, \ldots, T, \tag{5}$$

or in the absence of noise,

$$\min_{\mathbf{s}^t} \sum_{j} |s_j^t| \text{ subject to } \mathbf{A}s^t = \mathbf{x}^t \quad \text{for } t = 1, \ldots, T. \tag{6}$$

Expanding into positive and negative coefficients as in [5], the latter can be formulated as a linear programming problem for each t.

The mixing matrix $\mathbf{A}$ can either be estimated beforehand (as shown in Section 3), or by external optimization using

$$\min_{\mathbf{A}} \sum_{jt} |s_j^t(\mathbf{A})|, \tag{7}$$

where the $s_j^t(\mathbf{A})$'s represent, at each iteration, the solution of Eq. (6) under the current estimate of $\mathbf{A}$.

A similar two-step approach can be found in [8,9], where a learning rule for $\mathbf{A}$ is derived by fitting a multivariate Gaussian around the current estimate of the source components.

3. A potential-function-based method for estimating the mixing matrix and its implementation in the two-dimensional case

Following from Eq. (2), if only one of the sources (say, source i) was different from zero, then all $\mathbf{x}^t$'s would be proportional to $\mathbf{a}^i$ and all data points in mixture space would be aligned along the direction of this basis vector. When the sources are sparse, smaller coefficients are more likely and thus, for a given data point t, if one of the sources is significantly larger, the remaining ones are likely to be close to zero. Thus, the density of data in mixture space, besides decreasing with the distance from the origin, shows a clear tendency to *cluster* along the directions of the basis vectors $\mathbf{a}^j$'s. Estimating the mixing matrix, then, consists of finding the directions of maximum data density. For $M=2$, a simple and useful representation of mixture space is a scatter plot of the data, that shows x_2^t against x_1^t for every data point t. Fig. 1a, later in this section, shows an example of a scatter plot.

Our approach to estimating the mixing matrix consists of defining a local basis function around the neighbor directions of each data point, and then computing a potential function over all possible directions as the sum of the individual contributions. Local maxima of the potential function correspond then to the estimated directions of the basis vectors. This approach is developed here for the case $M=2$, when mixture space is a plane and directions can be parameterized using the angle θ in polar coordinates. Let $l_t=\sqrt{(x_1^t)^2+(x_2^t)^2}$ and $\theta_t=\tan^{-1}(x_2^t/x_1^t)$ be the radius and angle, respectively, of data point $\mathbf{x}^t$, and let α be the angular difference between an arbitrary direction and θ_t. We choose our *basis function* ϕ around $\mathbf{x}^t$ as a triangular function of the local angle α,

$$\phi(\alpha)=1-\frac{\alpha}{\pi/4} \quad \text{for } |\alpha|<\pi/4,$$
$$=0 \quad \text{elsewhere} \tag{8}$$

and we define a global *potential function* Φ over the absolute angle θ as

$$\Phi(\theta,\lambda)=\sum_t l_t\phi(\lambda(\theta-\theta_t)) \tag{9}$$

with λ a parameter to adjust the desired angular width or resolution of the local contributions, and l_t a weight to put more emphasis on the more reliable data (for equal perturbations, angular errors will be smaller for data points that are farther from the origin). In this work, the choice of the basis function ϕ and the setting of parameter λ were heuristical, as discussed in Section 6.

For practical purposes, the potential field is discretized by taking a sample of K points using an equally spaced grid over the semiplane $\theta_k=\pi/2K+k\pi/K$, $k=1,\ldots,K$, yielding $\Phi(\theta_k,\lambda)$. Local maxima of the resulting function are then identified as the columns of the estimated mixing matrix. Notice that with this approach, with a proper setting for λ, it is not necessary to know the *number* of sources beforehand, since it is *inferred* from the number of local maxima in the potential function. Fig. 1a shows the scatter plot of the data in the `FourVoices` example, described in Section 6, and Fig. 1b shows the potential function obtained with these data. The depicted stars show the original basis vectors of the mixing matrix, and the radial straight lines correspond to the inferred directions.

The computational complexity of this algorithm is $O(T\times K)$, to compute the angles between grid and data points. In practice, and without loss of performance, the number of data points T can be reduced significantly by discarding the less reliable ones, $l_t<h$, with threshold h adjusted experimentally (see Section 6). On the other hand, if the number K of grid points is too small, the sampling resolution $(180/2K)$ would be very poor (see Section 6 for a discussion). Yet, when high accuracy is required, one can start the computations with a coarser grid, and refine the results around the inferred directions either with a thinner grid or with continuous maximization. The optimization strategy of Section 2 is another way to get a higher accuracy.

Similar approaches to estimate the mixing matrix were described in [12] (using a histogram rather than a potential function) for the case $M=N=2$,

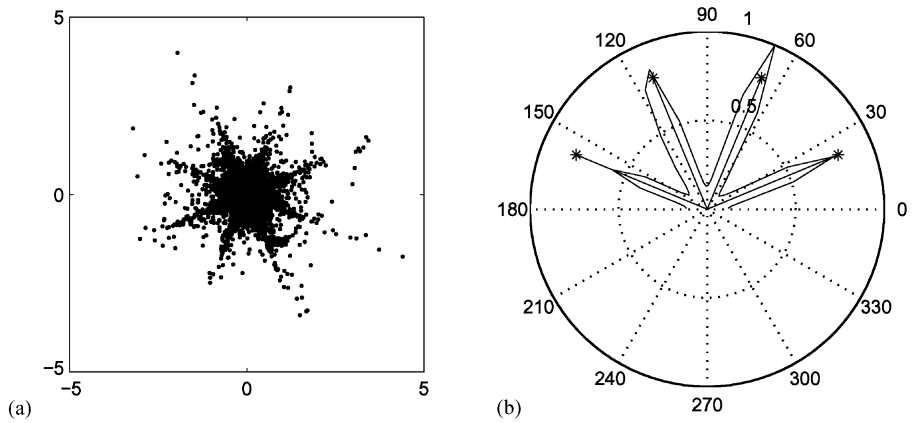

Fig. 1. (a) Scatter plot $\mathbf{X}_{2\cdot}$ vs. $\mathbf{X}_{1\cdot}$ of the data in the `FourVoices` example of Section 6. (b) Polar plot of the potential function $\Phi(\theta)$. Stars show the actual directions of the basis vectors and radial straight lines show the inferred ones.

and in [10] (using feature detection tools from image analysis).

4. l_1 Norm decomposition for the estimation of the sources, and its implementation in the two-dimensional case

Even when the mixing matrix A is known, since the system in Eq. (1) is underdetermined, its solution is not unique. The usual approach to sparse BSS consists of finding the solution that minimizes the l_1 norm, as in Eq. (6). In this case, the optimal representation of the data point

$$\mathbf{x}^t = \sum_j \mathbf{a}^j s_j^t$$

that minimizes $\sum_j |s_j|$ is the solution of the corresponding linear programming problem. Geometrically, for a given feasible solution, each source component is a segment of length $|s_j|$ in the direction of the corresponding $\mathbf{a}^j$ and, by concatenation, their sum defines a path from the origin to $\mathbf{x}^t$. Minimizing $\sum_j |s_j|$ amounts therefore to finding the *shortest path* to $\mathbf{x}^t$ over all feasible solutions. Notice that, with the exception of singularities, since mixture space is M-dimensional, M (independent) basis vectors $\mathbf{a}^j$ will be required for a solution to be feasible (i.e., to reach $\mathbf{x}^t$ without error).

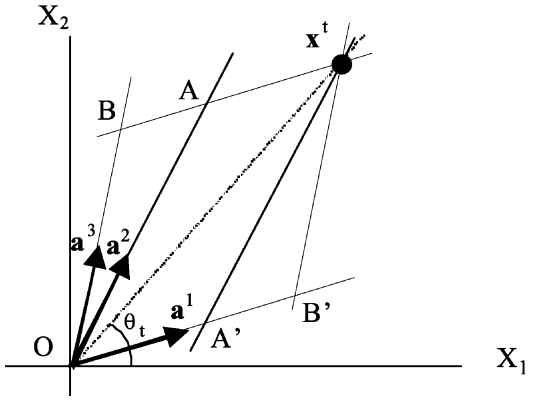

Fig. 2. The *shortest path* from the origin to the data point $\mathbf{x}^t$ is O–A–$\mathbf{x}^t$ (or O–A′–$\mathbf{x}^t$). Therefore, $\mathbf{x}^t$ decomposes as O–$\mathbf{A}'$ along direction $\mathbf{a}^1$ and as O–A along direction $\mathbf{a}^2$ (see text).

For the two-dimensional case (see Fig. 2), the shortest path is obtained by choosing the basis vectors $\mathbf{a}^b$ and $\mathbf{a}^a$ whose angles $\tan^{-1}(a_2^b/a_1^b)$ and $\tan^{-1}(a_2^a/a_1^a)$ are closest from below and from above, respectively, to the angle θ^t of $\mathbf{x}^t$. That is, the basis vectors that *enclose* $\mathbf{x}^t$.

Let $\mathbf{A}_r = [\mathbf{a}^b \mathbf{a}^a]$ be the *reduced* square matrix that includes only the selected basis vectors, let $\mathbf{W}_r = \mathbf{A}_r^{-1}$ and let $\mathbf{s}_r^t$ be the decomposition of the target point along $\mathbf{a}^b$ and $\mathbf{a}^a$. The components of the

sources are then obtained as

$$\mathbf{s}_r^t = \mathbf{W}_r \mathbf{x}^t,$$

$$s_j^t = 0 \quad \text{for } j \neq b, a. \tag{10}$$

In practice, when applied to all $t = 1, \ldots, T$, each reduced matrix $\mathbf{W}_r$ only needs to be computed once for all data points between any two pairs of basis vectors.

5. Sparsity and selection of the representation domain

Very often the data in the time domain do not satisfy the requirement of sparsity required for the above approach. For $N = M$ (square A) good results can sometimes still be found, as long as the scatter plot shows higher density in the directions of the basis vectors. However, in the underdetermined case higher sparsity is a requirement for good *separability* of the sources, even in the case when the mixing matrix is known.

In this situation a possible approach is to look for a linear transform T such that the new representation of the data is sparser. The transformation being linear, the mixing matrix is preserved and Eq. (1) can be rewritten as

$$T(\mathbf{X}) = \mathbf{A} T(\mathbf{S}). \tag{11}$$

The blind source separation, then, is performed *in the transformed domain*. This approach was proposed in [14] for the $N = M$ case and applied successfully to the separation of musical sources. The selected transform was an FFT-based spectrogram, and the inverse of the mixing matrix was estimated with better accuracy than similar methods in the time domain. Once the inverse mixing matrix was found, the sources were recovered in the time domain.

For $M < N$ the transform has to be *invertible*, so that the recovered sources $T(\mathbf{S})$ can be inverse-transformed back to the time domain. Thus, the procedures in Sections 3 and 4 apply directly to Eq. (11) simply by reinterpreting t as the appropriate index in the transformed domain (e.g., when T stands for the FFT transform, t will represent the discrete-frequency index).

The benefits of such an approach are clear in Fig. 3. Six flute signals playing different notes (see the SixFlutes example in Section 6) were synthetically mixed into two mixtures using equally spaced angles between the basis vectors. Fig. 3a presents a scatter plot of the resulting data (x_2^t against x_1^t for every t), showing a single big cloud. As can be seen, the different sources are indistinguishable. Then each mixture was FFT-transformed (see Section 6 for details) and the scatter plot of the frequency domain data is shown in Fig. 3b. The difference is extraordinary. Now almost all significant data points are neatly clustered along the six directions of the basis vectors, thus providing very good separability.

6. Experiments and results

6.1. Outline of the overall procedure

In order to prepare the mix, the following steps were followed:

Mixing

- In order to achieve a balanced mix, all sources were normalized to energy 1, $\mathbf{S}' = \mathbf{S}/\|\mathbf{S}\|$.
- A $2 \times N$ mixing matrix $\mathbf{A}$ was constructed by setting the basis vectors $\mathbf{a}^j$'s to unit length and (unless stated otherwise) equally spaced angles.
- The mixtures were obtained as in Eq. (1).
- The mixtures were rescaled to fill a $(-1,1)$ dynamic range, $\mathbf{X}' = \mathbf{X}/\max_{it}|x_i^t|$.

These mixtures were then used as input to the separation procedure, using the following steps:

Signal analysis

- The mixtures were processed in frames of length L samples and (unless stated otherwise) they were multiplied by a Hanning window. A "hop" distance d was used between the starting point of successive frames, leading to an overlap of $L - d$ samples between consecutive frames.
- Each frame was transformed with a standard FFT of length L, and the real and imaginary parts of the positive half spectrum were taken, for a total of L coefficients.
- For each mixture, the coefficients of successive frames were concatenated in a single vector, which was the actual input to the separation

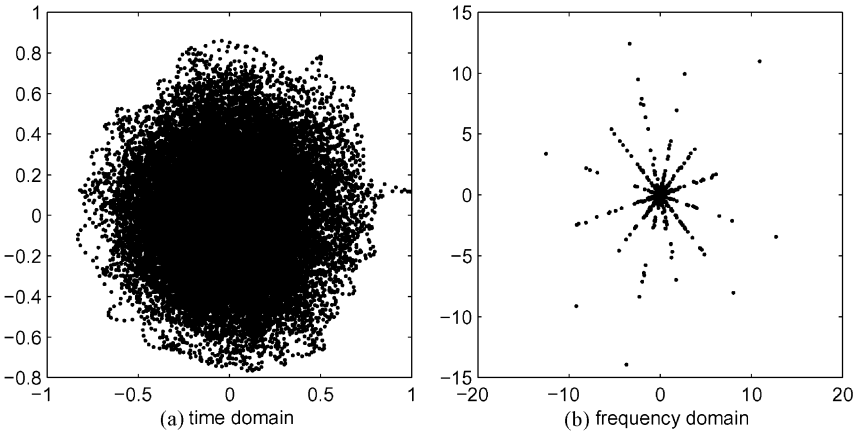

Fig. 3. Scatter plot $\mathbf{X}_2$. vs. $\mathbf{X}_1$. of six flute notes mixed into two mixtures with equally spaced angles in the (a) time and (b) frequency domains.

procedure. In terms of the previous sections, the set of all those frequency-domain coefficients plays the role of $\mathbf{X}$.

Source separation

- An estimate $\hat{\mathbf{A}}$ of the mixing matrix was found according to Section 5, using the basis function ϕ (8), the scaling parameter λ, the threshold h, and K grid points.
- Using $\hat{\mathbf{A}}$ (unless stated otherwise), an estimate of the sources was obtained following the procedure of Section 3 which, for a given set of data, has a unique solution.

Resynthesis

- For each estimated source, the coefficient vector was split back into frames.
- The real and imaginary components were regrouped into complex coefficients, and the spectra were extended to negative frequencies. For each spectrum, the standard IFFT transform was used to obtain time-domain frames of length L.
- Each frame was multiplied by the inverse window, and the overlap between frames was removed, 50% on either side, by keeping only the central part of the frame (thus avoiding the distortion at the edges that often appears after frequency-domain manipulation). The resynthetized signals were finally built by simple concatenation of the resulting pieces.

Finally, the quality of the separation was tested using the following measures:

Performance measures

- The error in the estimated matrix was measured by the difference between the angles of the estimated and the actual basis vectors (columns of the mixing matrix).
- The estimated sources $\mathbf{S}_j$ were rescaled to the same energy level as their corresponding original sources.
- A *reconstruction index* was defined as a signal-to-noise ratio of the error, that is

$$S/N = 10 \log \frac{||\hat{\mathbf{S}} - \mathbf{S}||^2}{||\mathbf{S}||^2}. \tag{12}$$

6.2. Parameter setting

The setting of the different parameters was done heuristically by optimizing the overall performance of the algorithm. For the data sets that were used, the Hanning window performed better than a Hamming window or a square window. The length L of the frame was selected among a few consecutive powers of 2 and the hop distance d was set around $0.3L$ to provide enough overlap without overloading the system with data.

The triangular basis function ϕ was compared with a round and a square basis functions, with little differences, but ϕ seemed to provide better resolution. The threshold h was set to 0.3, about a third of the dynamic range. The most sensitive

parameter was λ because the number of maxima of the potential function (i.e., the estimated number of sources) depended on it. As it is often the case with clustering algorithms, the proper choice of λ depends on the data (the actual quality of the clusters). Thus, for each experiment below, we simply evaluated the range $[\lambda_{\min}, \lambda_{\max}]$ over which the estimated number of sources were correct, and kept all further experiments within that range. Finally, unless stated otherwise, the number K of grid points for the potential function was set so that a grid point was coaligned with (had the same angle as) each of the basis vectors $\mathbf{a}^j$ (thus allowing for a perfect estimate), and so that there was at least one grid point between every two consecutive vectors (since at least one sampling point is needed to provide a minimum between two adjacent local maxima).

6.3. Experiments with steady sources

The approach was first tested using the SixFlutes data set: the sound of a flute playing steady, isolated notes was recorded at high quality in an acoustically isolated booth without reverberation, and sampled at 44.1 kHz with 16 bits resolution[4] Six 743 ms excerpts (32768 samples) were selected for the sources, corresponding to the notes a4, d5, f5, g5, c6 and d#6. These sounds were so steady (the spectra varied so little over time) that the whole signal could be processed with a single FFT with $L = 32768$, thus avoiding the use of frames or windowing. In experiment SixFlutes I, the clustering algorithm of Section 3 was run with parameter $\lambda = 5$ and a grid with $K = 30$ equally spaced samples. The clustering was successful and the mixing matrix was recovered exactly. A maximally sparse estimation of the sources was then obtained with the separation procedure of Section 4. The reconstruction indices obtained are shown in Table 1. When listening to the recovered signals the correct notes were very clear, but a little background noise was present (the accumulated sounds of the player blowing into the flute, plus some traces of cross-talk). Similar results were obtained with the external optimization approach of Section 2 (Eq. (7)) when the starting state was not too far from the solution (otherwise it got trapped in local minima), but this procedure was much slower.

The experiment was then repeated several times using random mixing matrices. The matrix was always correctly estimated within the 3 degrees of resolution provided by the grid, but the reconstruction indices dropped. The following two experiments were then devised to measure the sensitivity of the separation algorithm to the accuracy in the estimation of the matrix and to the closeness of the sources, independently. Experiment SixFlutes II was identical to SixFlutes I except for the grid points, which were shifted 3° from their original positions and therefore were no longer aligned with the basis vectors. After clustering, the estimated angles were all off by 3°, as expected. Results of the separation (Table 1) were impaired by 8.1 dB on average. In experiment SixFlutes III the basis vectors were lumped together in a total span of 6°, so that each source was separated by only 1° from the next. The number of grid points was set to $K = 540$ so as to guarantee the alignment, and $\lambda = 55$ was required to get enough resolution. With this setting, the mixing matrix was again perfectly recovered and separation indices are shown in Table 1. The loss was now only 4.1 dB on average with respect to the SixFlutes I experiment, which illustrates the relative insensitivity of the separation procedure to the proximity between the sources.

For the sake of comparison, the last experiment (SixFlutes IV) was conducted on the same data set using the mixtures in the time domain instead of the frequency domain. The maxima of the potential function were no longer in the directions of the basis vectors and therefore the estimate of the matrix was meaningless. The separation was then attempted using the original mixing matrix instead, but the algorithm still failed to separate the sources, as shown in Table 1.

For the first three experiments above the operative range for parameter λ was found to be $[0.9, 209.4]$, $[0.0, 44.1]$ and $[51.1, 3462.0]$, respectively. For SixFlutes IV the algorithm failed no matter what the value of λ was.

[4] All flute examples were performed by Linda Antas, University of Washington.

Table 1
S/N reconstruction indices (dB) for the different experiments (see text)

SixFlutes I	50.5	52.5	49.4	43.4	49.1	51.8
SixFlutes II	41.2	36.0	50.8	41.7	35.6	42.5
SixFlutes III	47.7	42.8	43.3	37.2	47.2	54.0
SixFlutes IV	−1.9	−2.0	−2.2	−2.4	−2.3	−2.4
FourVoices	21.7	19.4	15.7	16.6		
FiveSongs	15.6	15.5	15.0	15.1	15.2	
SixFluteMelodies	20.4	19.4	14.2	16.1	24.7	29.1

6.4. Experiments with dynamic sources

The three experiments presented next were performed on much more dynamic signals, and the frame-by-frame analysis described above was required. The experiments were conducted on the following sets of signals: A FourVoices data set with four 2.9 s sentences pronounced by four different people (three females and a male), recorded at 22,050 Hz and 8 bits with a low-quality microphone on a home personal computer. Pre-processing was done with $L = 2048$ and $d = 614$ samples. A FiveSongs data set with five 5 s long full-ensemble music pieces (two classical and three pop/folk music) extracted from standard CDs (44,100 Hz/16 bits), downsampled to 11,025 Hz monophonic and processed with $L = 4096$ and $d = 1228$ samples. Finally, a SixFluteMelodies data set (see footnote 4) including six 5.7 s long flute melodies (the two voices of a canon, the two voices of a duet and two unrelated melodies) with a high-quality registration at 44,100 Hz/16 bits, down-sampled to 22,050 Hz and processed with $L = 8192$ and $d = 3276$ samples.

In all three cases the mixing matrix was formed with equally spaced angles, and the number of grid points was selected for perfect alignment ($K = 36, 35$, and 30, respectively) in order to be able to measure the maximum separation ability of the system. As the SixFlutes experiments had shown, the estimation of the mixing matrix was always successful, and the operative range for λ laid in the intervals $[1.4, 94.2]$, $[0.1, 14.6]$, and $[1.6, 145.0]$, respectively. Results of the separation are shown in Table 1. Although good enough in themselves, the reconstruction indices of the dynamic signals were significantly poorer than those of the SixFlutes I experiment, in part due to the intrinsic difficulties of the short-term analysis and resynthesis. Reconstruction indices were on the same range for the three examples, regardless of the number of voices, with somehow worse results in the case of the FiveSongs, probably due to the higher complexity of the sounds. The plot of the recovered signals was, in all cases, very similar to the plot of the original sources, as illustrated in Fig. 4 for the FourVoices case. From a subjective listening point of view, the separation of the FourVoices example was remarkable for the high intelligibility of the recovered sentences, in spite of some background noise and cross-talk. In the case of the FiveSongs, the reconstructed songs were also very clear but the quality of the sound was sensibly degraded by background noise, cross-talk and a flattening of percussive sounds and sharp transitions. Finally, in the SixFluteMelodies example, although the recovered melodies were clear, a sort of ringing artifact appeared in the transitions between notes, and some frame-rate rattling noise was present. Sound examples for the above experiments are available on-line in [3].

7. Discussion and further work

In the context of underdetermined blind source separation (i.e., BSS with fewer mixtures than sources), the three main contributions of this paper have been the benefits of performing blind source separation in the frequency domain (rather than in the time domain); a clustering algorithm for the estimation of the mixing matrix in the two-sensor case; and a shortest path separation procedure that yields the most sparse estimate of the sources from

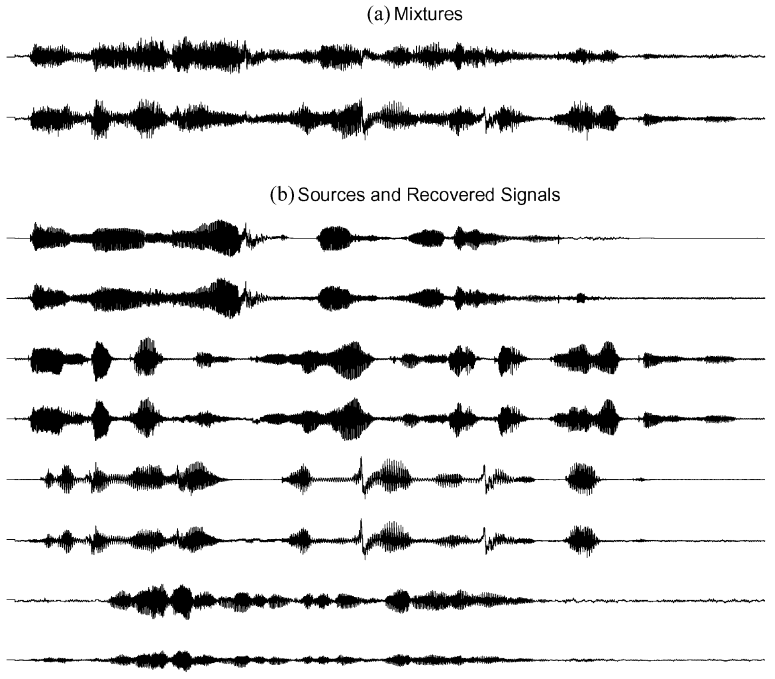

Fig. 4. FourVoices experiment: (a) mixtures and (b) sources and recovered signals (pairwise).

two mixtures. Several experiments have been presented involving music and speech signals, with rather good results, including the successful separation of six sources from two mixtures.

From the above three points the most effective contribution to a successful separation has probably been the exploitation of sparsity in the frequency domain since, as experiments have shown, only the transformed data satisfy the assumptions of sparsity required by the clustering and separation algorithms. The reason for this is probably the highly harmonic nature of speech and music signals. With a proper setting for λ, the estimation of the mixing matrix has always been successful, within the accuracy provided by the sampling grid, and the separation was more adversely affected by an inaccurate estimate of the basis vectors than by the proximity of the mixed sources to each other. S/N reconstruction indices have shown excellent scores for steady flute notes that could be processed with a single FFT, and good scores for the other three dynamic examples, which required short-term analysis and resynthesis. The recovered signals have been highly intelligible to the ear in all cases, in spite of some background noise and some cross-talk. Results seem to show that the difficulty of the separation depends more on the complexity of the sounds than on the number of sources present, but further experiments would be required in order to assess this trend.

Even if l_1 norm minimization is theoretically the most likely a posteriori estimation for Laplacian sources, in practice good separation is obtained only where the sources are *disjoint* or almost disjoint, regardless of whether they are Laplacian or not. This is usually the case for the overtones of signals with different pitch, for instance. But when sources overlap, the shortest path separation criterion, although statistically optimal, is unable to give the credit to the sources actually involved.

The main goal of this work has been the validation of the overall procedure, but many particular aspects require further study. The setting of the parameters was heuristical. Little exploration was done for the analysis and resynthesis procedure (for instance, a 50% overlap with a triangular

window would probably improve the resynthesis). A deeper study of the distribution of the sources (both in the time and frequency domain) would be useful, and other representation domains could be evaluated (Gabor, wavelet, or even combined representations), that might be better adapted to transitions, or lead to improved sparsity (and hopefully disjointness) of the sources. Finally, the performance measures could be extended to include the cross-coherence of the reconstructed sources.

The work presented here can be extended to any number of sensors by using a clustering algorithm in the estimate of the mixing matrix, and standard linear programming for the decomposition into sources. Current and further work include the study of different decomposition criteria for the separation of synthetic signals with different degrees of sparsity [13], the evaluation of alternative analysis and resynthesis procedures, the study of other representation domains, and the extension of the overall procedure to delayed [2] and convolved mixtures.

References

[1] A.J. Bell, T.J. Sejnowski, An information-maximization approach to blind separation and blind deconvolution, Neural Comput. 7 (6) (1995) 1129–1159, http://www.cnl.salk.edu/cgi-bin/pub-search#articles.

[2] P. Bofill, Underdetermined blind separation of delayed sound sources in the frequency domain, Technical Report UPC-DAC-2001-14, http://www.ac.upc.es/homes/pau/.

[3] P. Bofill, M. Zibulevsky, Sound Examples, http://www.ac.upc.es/homes/pau/.

[4] O. Bermond, J.F. Cardoso, Méthodes de séparation de sources dans le cas sous-déterminé, Proceedings of the GRETSI'99, Vannes, France, 1999, pp. 749–752, http://www.tsi.enst.fr/ cardoso/jfbib.html.

[5] S.S. Chen, D.L. Donoho, A. Saunders, Atomic decomposition by basis pursuit, Technical Report, http://www-stat.stanford.edu/~donoho/Reports/.

[6] A. Hyvärinen, Survey on independent component analysis, Neural Comput. Surveys (2) (1999) 94–128, http://www.cis.hut.fi/projects/ica/.

[7] C. Jutten, J. Herault, Blind separation of sources, an adaptive algorithm based on neuromimetic architecture, Signal Process. 24 (1) (1991) 1–10.

[8] T.-W. Lee, M.S. Lewicki, M. Girolami, T.J. Sejnowski, Blind source separation of more sources than mixtures using overcomplete representations, IEEE Signal Process. Lett. 6 (4) (1999) 87–90.

[9] M.S. Lewicki, T.J. Sejnowski, Learning overcomplete representations, Neural Comput. 1998, in press, http://www.cs.cmu.edu/~lewicki.

[10] J.K. Lin, G. Grier, Faithful representation of separable distributions, Neural Comput. 9 (1997) 1305–1320.

[11] B.A. Olshausen, D.J. Field, Sparse coding with an overcomplete basis set: a strategy employed by V1?, Vision Res. 37 (1997) 3311–3325.

[12] A. Prieto, B. Prieto, C.G. Puntonet, A. Cañas, P. Martín-Smith, Geometric separation of linear mixtures of sources: application to speech signals, Proceedings of the ICA'99, January 1999, pp. 295–300, http://atc.ugr.es/~bprieto/sfuentes/articulo.html.

[13] L. Vielva, A. Subinas, E. Navas, I. Hernáz, P. Bofill, "Separación ciega de fuentes: caso indeterminado", submitted to URSI'01.

[14] M. Zibulevsky, B.A. Pearlmutter, Blind source separation by sparse decomposition, Technical Report No. CS99-1, University of New Mexico, Albuquerque, July 1999, http://www.cs.unm.edu/~bap/papers/sparse-ica-99a.ps.gz.