Maximum Likelihood Blind Image Separation Using Nonsymmetrical Half-Plane Markov Random Fields

Rima Guidara, Shahram Hosseini, and Yannick Deville, *Member, IEEE*

***Abstract*—This paper presents a maximum likelihood approach for blindly separating linear instantaneous mixtures of images. The spatial autocorrelation within each image is described using nonsymmetrical half-plane (NSHP) Markov random fields in order to simplify the joint probability density functions of the source images. A first implementation assuming stationary sources is presented. It is then extended to a more realistic nonstationary image model: two approaches, respectively based on blocking and kernel smoothing, are proposed to cope with the nonstationarity of the images. The estimation of the mixing matrix is performed using an iterative equivariant version of the Newton-Raphson algorithm. Moreover, score functions, required for the computation of the updating rule, are approximated at each iteration by parametric polynomial estimators. Results achieved with artificial mixtures of both artificial and real-world images, including an astrophysical application, clearly prove the high performance of our methods, as compared to classical algorithms.**

***Index Terms*—Blind source separation (BSS), maximum likelihood approach, nonstationary sources, nonsymmetrical half-plane (NSHP) Markov random fields.**

I. Introduction

BLIND source separation (BSS) methods aim at recovering a set of unobserved source signals from several observations which are the results of some unknown transformation of these sources. Since this problem is encountered in various fields, the interest for BSS techniques has increased in the last two decades, yielding numerous approaches which usually make some hypotheses on the sources and their transformation, in order to deal with this ill-posed problem. Many of these approaches especially assume that the sources are mutually statistically independent, and so apply an independent component analysis (ICA) [1] to recover the original signals.

Basic ICA methods also suppose each source is independently and identically distributed (i.i.d) and exploit the assumed source non-Gaussianity to perform separation [1], [2]. Obviously, these approaches cannot be applied to Gaussian source signals, and completely ignore some important features that can dramatically improve source estimation accuracy, for example, the temporal (*resp.* spatial) structure of the signals (*resp.* images). Indeed, it has been proven that the ICA problem may be solved by exploiting non-Gaussianity, autocorrelation or nonstationarity of the sources,[1] [4]. Although there are several methods in the literature that take into account either source autocorrelation [5]–[10] or source nonstationarity, e.g., [11]–[17], most of the proposed approaches only exploit part of the available information.

In [18], Hosseini *et al.* proposed a linear instantaneous BSS method based on a maximum likelihood (ML) approach, where *autocorrelated 1-D stationary* sources were modeled as qth-order Markov sequences. This method provides an asymptotically efficient estimator of the mixing matrix and takes into account both non-Gaussianity and temporal autocorrelation of the mutually independent sources. Nevertheless, it suffers from severe limitations which can cause, in some cases, the failure of the separation procedure. Indeed, this method is based on the source stationarity assumption which is actually unsatisfied for most real signals. Moreover, it only applies to 1-D time series and the computational load of the resulting algorithm critically increases when rising the number of samples or the order of Markov models in the source signals.

First inspired from this previous work, we develop in this paper a new blind image separation method based on an ML approach, where spatial interactions between neighboring pixels are modeled using Markov random fields (MRFs). Unlike [18], the proposed method is especially adapted to image separation. Furthermore, major modifications are introduced, in comparison to the initial approach, in order to reduce the computational cost and to adapt the method to nonstationary signals.

Since an initial paper by Geman and Geman [19], MRFs have been extensively used for modeling the spatial dependence between neighboring pixels within an image. Applications hence cover a wide range of fields, including for instance image segmentation, texture classification and feature extraction. In blind image separation, however, only a few reported BSS methods take advantage of the image autocorrelation by describing sources with MRF models [20]–[22]. These approaches still assume some prior knowledge about the conditional probablility density function (pdf) of the data, for instance a Gibbs prior [19], [23], which then allows one writing this conditional pdf in a closed form. In the present work, nevertheless, the Markov model is only used to provide a qualitative description of the statistical dependence between pixels. Therefore, we do not require any prior information about the source image distributions, so that the model may adapt to most real-life images.

Manuscript received July 03, 2008; revised May 01, 2009. First published July 10, 2009; current version published October 16, 2009. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Ilya Pollak.

The authors are with the Laboratoire d'Astrophysique de Toulouse-Tarbes, Université de Toulouse-CNRS, 31400 Toulouse, France (e-mail: rguidara@ast.obs-mip.fr; shosseini@ast.obs-mip.fr; ydeville@ast.obs-mip.fr).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TIP.2009.2027367

[1]Autocorrelation and nonstationarity can also be used to override the independence assumption (see, e.g., [3]).

Furthermore, we only focus our study on causal bidimensional MRF models, which are mainly attractive for their simplicity and computational efficiency. In particular, two types of causal MRF models are widely known in the literature: the Markov mesh random fields (MMRFs) [24]–[26] and the nonsymmetrical half-plane (NSHP) Markov random fields, also known as unilateral MRFs [27]–[29]. A comparison between both models in [30] proved that MMRFs may be classified as a subset of NSHP-MRF models, and recommended using NSHP-MRFs when accurate image processing is needed, especially due to their higher ability to capture the diagonal edge description in images. Therefore, we adopt an NSHP Markov model to describe autocorrelation within the source images in this paper.

The first method presented in this paper supposes that the sources are *stationary*. An extension to possibly *nonstationary* images is then proposed and two approaches, respectively based on blocking and kernel smoothing, are proposed to deal with the nonstationarity of the sources. This improved version operates in a quasi-optimal manner since it simultaneously exploits the non-Gaussianity, the autocorrelation and the nonstationarity of the images. Furthermore, the proposed method is able to take advantage of pdf nonstationarities, while classical nonstationary BSS methods only consider the nonstationarity of variance.

The remainder of this paper is organized as follows. In Section II, we present the BSS problem, then propose an ML separation method. The sources are described using NSHP-MRF models to simplify the log-likelihood function. We then define conditional score functions and derive the maximum of the log-likelihood, which yields the estimating equations. Finally, we propose a modified equivariant Newton-Raphson algorithm to solve these equations. In Section III, we introduce methods for estimating the conditional score functions, using parametric polynomial estimators. A simplified version of this approach assuming that the sources are stationary is first presented and extensions to the possibly nonstationary case are then proposed. In Section IV, we summarize the main steps of our algorithm. Experimental results using artificial mixtures of both artificial and real-world images are presented in Section V and compared to classical blind image separation algorithms. An application of the developed methods to Astrophysics is also presented. Conclusion and some prospects are finally presented in Section VI.[2]

II. Markovian Image Separation Method

We consider the blind image separation problem in its simplest form, where the observations are linear instantaneous mixtures of the source images. Assume we have K linear transformations of K source images composed of $N = N_1 \times N_2$ pixels. The mixture model in a noiseless context is defined by

$$\mathbf{x}(n_1, n_2) = \mathbf{A}\mathbf{s}(n_1, n_2) \tag{1}$$

where $\{n_1, n_2\} \in [1 \dots N_1] \times [1 \dots N_2]$, $\mathbf{x}(n_1, n_2)$ and $\mathbf{s}(n_1, n_2)$ are, respectively, the K-dimensional observation and source vectors, and $\mathbf{A}$ is an unknown $K \times K$ invertible mixing matrix. Our aim is then to estimate a separating matrix $\mathbf{B} = \mathbf{A}^{-1}$ up to a diagonal matrix and a permutation matrix, which are classical but, in most cases, not critical indeterminacies for BSS methods [1].[3]

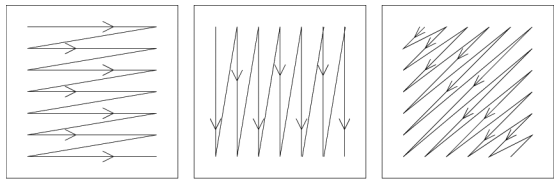

Fig. 1. Different sweeping schemes over an image: (a) Horizontal TV scanning, (b) vertical scanning, (c) diagonal scanning.

In this paper, we develop an ML approach for estimating this matrix. We, therefore, want to maximize, with respect to the separating matrix $\mathbf{B}$, the joint pdf of all the pixels of the images in the observation vector $\mathbf{x}$, defined by

$$\mathrm{F} = f_{\mathbf{x}}(x_1(1,1), \dots, x_K(1,1), \dots , x_1(N_1, N_2), \dots, x_K(N_1, N_2)). \tag{2}$$

Supposing that the source images are mutually independent, this joint pdf may be rewritten as

$$\mathrm{F} = \left(\frac{1}{|\det(\mathbf{B}^{-1})|}\right)^N \prod_{i=1}^{K} f_{\mathbf{b}_i^T\mathbf{x}} (\mathbf{b}_i^T\mathbf{x}(1,1), \dots, \mathbf{b}_i^T\mathbf{x}(N_1, N_2)) \tag{3}$$

where $\mathbf{b}_i^T$ is the i-th row of $\mathbf{B}$ and $f_{\mathbf{b}_i^T\mathbf{x}}(\mathbf{b}_i^T\mathbf{x}(1,1), \dots, \mathbf{b}_i^T\mathbf{x}(N_1, N_2))$ represents the joint pdf of the N pixels of $\mathbf{b}_i^T\mathbf{x}$. Rewriting the above equation as a function of the sources $s_i = \mathbf{b}_i^T\mathbf{x}$, $\forall i = 1, \dots, K$, we obtain

$$\mathrm{F} = \left(\frac{1}{|\det(\mathbf{B}^{-1})|}\right)^N \prod_{i=1}^{K} f_{s_i}(s_i(1,1), \dots, s_i(N_1, N_2)) \tag{4}$$

where $f_{s_i}(s_i(1,1), \dots, s_i(N_1, N_2))$ represents the joint pdf of the N pixels of source s_i.

Using Bayes rule, we want to decompose this joint pdf and then simplify it by assuming MRF models for the source images.

A. Bidimensional Markov Model

Contrary to temporal signals, where the natural order for Bayes decomposition just follows time evolution, there exists no natural order for the pixels in an image, so that different schemes my be used to perform this decomposition. Nevertheless, it is obviously more interesting to preserve continuity when decomposing the joint pdf, which makes it possible to handle the local interactions between pixels. Some particular almost continuous sweeping schemes over an image are shown in Fig. 1.

These sweeping trajectories being almost equivalent for our approach, we choose the first one, which is a classical horizontal

[2]Some of the results developed here have been partly presented in our conference papers [31]–[35].

[3]The scaling and permutation indeterminacies may be critical in some applications. This is the case, for example, in some applications related to hyperspectral image separation [21], [36].

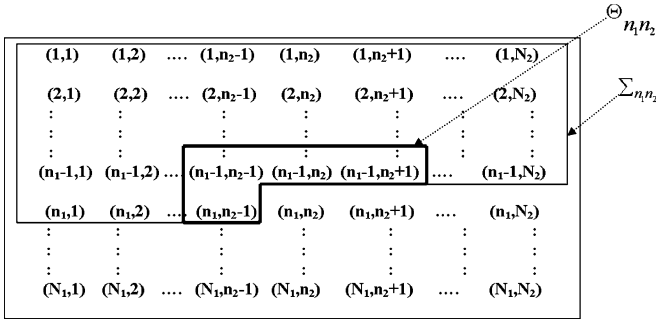

Fig. 2. Sets defined for a NSHP Markov random field: $\Sigma_{n_1n_2}$ is the NSHP associated to the site (n_1, n_2), $\Theta_{n_1n_2}$ is the support of (n_1, n_2) which defines the order of the NSHP Markov model.

TV scanning order. Each source joint pdf f_{s_i} in (4) can then be decomposed using Bayes rule as

$$\begin{aligned} &f_{s_i(1,1)}(s_i(1,1)) \times f_{s_i(1,2)}(s_i(1,2)|s_i(1,1)) \times \cdots \\ &\quad \times f_{s_i(1,N_2)}(s_i(1,N_2)|s_i(1,N_2-1),\ldots,s_i(1,1)) \\ &\quad \times f_{s_i(2,1)}(s_i(2,1)|s_i(1,N_2),\ldots,s_i(1,1)) \times \cdots \\ &\quad \times f_{s_i(2,N_2)}(s_i(2,N_2)|s_i(2,N_2-1),\ldots,s_i(1,1)) \cdots \\ &\quad \times f_{s_i(N_1,N_2)}(s_i(N_1,N_2)|s_i(N_1,N_2-1),\ldots,s_i(1,1)) \quad (5) \end{aligned}$$

where each factor represents the conditional pdf of a pixel with respect to its predecessors, according to the chosen sweeping scheme. This formulation is particularly interesting since, to each pixel $s_i(n_1, n_2)$ corresponds a set of predecessors located in the nonsymmetrical half-plane (NSHP) defined by

$$\Sigma_{n_1n_2} = \{(k,l) \in [1 \ldots N_1] \times [1 \ldots N_2] / (k = n_1, l < n_2) \cup (k < n_1)\}. \quad (6)$$

Using the decomposition (5) to rewrite the joint pdf $f_{s_i}(s_i(1,1),\ldots,s_i(N_1,N_2))$, we obtain the general form

$$f_{s_i}(s_i(1,1),\ldots,s_i(N_1,N_2)) = \prod_{n_1=1}^{N_1} \prod_{n_2=1}^{N_2} f_{s_i(n_1,n_2)}(s_i(n_1,n_2)|s_{i_{\Sigma_{n_1n_2}}}) \quad (7)$$

where $s_{i_{\Sigma_{n_1n_2}}}$ is the set of pixel values located in the NSHP $\Sigma_{n_1n_2}$, and $f_{s_i(n_1,n_2)}(s_i(n_1,n_2)|s_{i_{\Sigma_{n_1n_2}}})$ is the conditional pdf of the pixel $s_i(n_1, n_2)$, with respect to $s_{i_{\Sigma_{n_1n_2}}}$.

To simplify this function, we suppose that each source image can be modeled by an NSHP-MRF [27]–[29], defined as follows.

Definition: Let s be a random field defined over an $N_1 \times N_2$ rectangular lattice L. For each couple of coordinates (n_1, n_2), called a site of the lattice, we define an NSHP $\Sigma_{n_1n_2}$ and a support $\Theta_{n_1n_2} \subset \Sigma_{n_1n_2}$ in this lattice (see Fig. 2 and (10) below for an example). s is called an NSHP Markov random field (or a unilateral Markov random field) if and only if its density f is such that

$$f(s(n_1,n_2)|s_{\Sigma_{n_1n_2}}) = f(s(n_1,n_2)|s_{\Theta_{n_1n_2}}) \forall (n_1,n_2) \in L \quad (8)$$

where $s_{\Theta_{n_1n_2}}$ is defined as $s_{i_{\Sigma_{n_1n_2}}}$ above.

For simplicity, the support set $\Theta_{n_1n_2}$ is generally chosen so that it has the same shape for any site (n_1, n_2) of the lattice. Identifying each source image with an $N_1 \times N_2$ lattice, we here suppose that the interactions between pixels are described by an NSHP Markov random field according to the above definition. The joint pdf of each image can then be written in the following form:

$$f_{s_i}(s_i(1,1),\ldots,s_i(N_1,N_2)) = \prod_{n_1=1}^{N_1} \prod_{n_2=1}^{N_2} f_{s_i(n_1,n_2)}\left(s_i(n_1,n_2)|s_{i_{\Theta_{n_1n_2}}}\right). \quad (9)$$

We here assume that the sources are fourth-order NSHP Markov fields,[4] since for each pixel $s(n_1, n_2)$, we define the support set as

$$\Theta_{n_1n_2} = \Big\{(n_1, n_2-1), (n_1-1, n_2+1) (n_1-1, n_2), (n_1-1, n_2-1)\Big\}. \quad (10)$$

This type of support is attractive since it provides a sufficient description of the dependence between neighboring pixels in the horizontal, vertical and even in the two diagonal directions. Moreover, it is proved in [29] that a fourth-order NSHP Markov model takes into account the autocorrelation between the pixel and all of its neighboring pixels (see Appendix A).

B. Estimating Equations

Under the fourth-order NSHP Markov model assumption, the conditional pdf of each pixel given all its predecessors is equal to its conditional pdf given only its three top neighbors and its left neighbor. This condition is obviously satisfied anywhere in the image except for some pixels situated on the left, top and right boundaries. However, given a quite large number of pixels $N = N_1 \times N_2$, the conditional pdf of the boundary pixels may be neglected without significant loss in estimation performance. The joint pdf (9) of each source image may be written in this case as

$$f_{s_i}(s_i(1,1),\ldots,s_i(N_1,N_2)) \simeq \prod_{n_1=2}^{N_1} \prod_{n_2=2}^{N_2-1} f_{s_i(n_1,n_2)}\left(s_i(n_1,n_2)|s_{i_{\Theta_{n_1n_2}}}\right) \quad (11)$$

where $\Theta_{n_1n_2}$ is defined in (10) and $f_{s_i(n_1,n_2)}(s_i(n_1,n_2)|s_{i_{\Theta_{n_1n_2}}})$ is the conditional pdf of the pixel $s_i(n_1, n_2)$ with respect to the pixels located in the support set $\Theta_{n_1n_2}$. Taking the logarithm of (4) and replacing each source pdf by the approximation in (11), the log-likelihood function takes the form

$$\log \mathrm{F} \simeq N \log(|\det(\mathbf{B})|) + \sum_{i=1}^{K} \sum_{n_1=2}^{N_1} \sum_{n_2=2}^{N_2-1} \log f_{s_i(n_1,n_2)}\left(s_i(n_1,n_2)|s_{i_{\Theta_{n_1n_2}}}\right). \quad (12)$$

[4]The order of the NSHP Markov random field here corresponds to the number of pixels in the support pixel set $\Theta_{n_1n_2}$, following the definition in [27]. In the literature, the orders of noncausal bidimensional MRFs are generally associated to the distance between pixels [23]. Using the latter convention, our MRF model is a second-order bidimensional MRF.

Dividing the above function by N and defining the spatial average operator[5] $E_N[.] = 1/N \sum_{n_1=2}^{N_1} \sum_{n_2=2}^{N_2-1}[.]$, we obtain a simplified expression of the log-likelihood, written as

$$L = \frac{1}{N}\log \mathrm{F} \simeq \log(|\det(\mathbf{B})|) + E_N\left[\sum_{i=1}^{K} \log f_{s_i(n_1,n_2)}\left(s_i(n_1,n_2)|s_{i_{\Theta_{n_1 n_2}}}\right)\right]. \quad (13)$$

We aim at maximizing this cost function with respect to the separating matrix $\mathbf{B}$. This can be done, for example, using a relative gradient ascent algorithm as in [18]. Nevertheless, this approach is very time consuming, especially for images with a large number of pixels. Thus, in the following section, we propose to use a modified Newton-Raphson algorithm to solve this problem. This algorithm requires us to compute the gradient of (13), which reads, using (10)

$$\frac{\partial L}{\partial \mathbf{B}} \simeq \mathbf{B}^{-T} + E_N\left[\frac{\partial}{\partial \mathbf{B}} \sum_{i=1}^{K} \log f_{s_i(n_1,n_2)}(s_i(n_1,n_2)| s_i(n_1,n_2-1), s_i(n_1-1,n_2+1) \; s_i(n_1-1,n_2), s_i(n_1-1,n_2-1))\right]. \quad (14)$$

Defining the conditional score function of source s_i with respect to $s_i(n_1-k,n_2-l)$ by

$$\psi^{k,l}_{s_i(n_1,n_2)}(n_1,n_2) = \frac{-\partial}{\partial s_i(n_1-k,n_2-l)} \log f_{s_i(n_1,n_2)} (s_i(n_1,n_2)|s_i(n_1,n_2-1), s_i(n_1-1,n_2+1) \; s_i(n_1-1,n_2), s_i(n_1-1,n_2-1)) \quad (15)$$

and using the chain rule, we obtain

$$\frac{\partial L}{\partial \mathbf{B}} \simeq \mathbf{B}^{-T} - \sum_{i=1}^{K} E_N\left[\sum_{(k,l)\in\Upsilon} \psi^{k,l}_{s_i(n_1,n_2)}(n_1,n_2) . \frac{\partial s_i(n_1-k,n_2-l)}{\partial \mathbf{B}}\right] \quad (16)$$

where $\Upsilon = \{(0,0),(0,1),(1,-1),(1,0),(1,1)\}$.

The pixel $s_i(n_1-k,n_2-l)$ reads as a function of $\mathbf{B}$ and the observations as

$$s_i(n_1-k,n_2-l) = \mathbf{b}_i^T \mathbf{x}(n_1-k,n_2-l) \quad (17)$$

where $\mathbf{b}_i^T$ is the i-th row of $\mathbf{B}$. This yields

$$\frac{\partial s_i(n_1-k,n_2-l)}{\partial \mathbf{b}_j^T} = \mathbf{x}^T(n_1-k,n_2-l)\delta_{ij} \quad (18)$$

where δ_{ij} is the Kronecker symbol defined by

$$\delta_{ij} = \begin{cases} 1, & \text{if } i=j \\ 0, & \text{otherwise.} \end{cases}$$

[5]The boundary pixels have been neglected again in this spatial average.

Then, we can see that $\psi^{k,l}_{s_i(n_1,n_2)}(n_1,n_2).\partial s_i(n_1-k,n_2-l)/\partial \mathbf{B}$ is a matrix whose i-th row is equal to $\psi^{k,l}_{s_i(n_1,n_2)}(n_1,n_2).\mathbf{x}^T(n_1-k,n_2-l)$ while all its other rows are zero, so that

$$\sum_{i=1}^{K}\left[\psi^{k,l}_{s_i(n_1,n_2)}(n_1,n_2).\frac{\partial s_i(n_1-k,n_2-l)}{\partial \mathbf{B}}\right] = \mathbf{\Psi}^{k,l}_{\mathbf{s}}(n_1,n_2)\mathbf{x}^T(n_1-k,n_2-l) \quad (19)$$

where $\mathbf{\Psi}^{k,l}_{\mathbf{s}}(n_1,n_2)$ is a K-dimensional column vector containing all score functions $\psi^{k,l}_{s_i(n_1,n_2)}(n_1,n_2)$. Inserting this equation in (16), we obtain

$$\frac{\partial L}{\partial \mathbf{B}} \simeq \mathbf{B}^{-T} - E_N\left[\sum_{(k,l)\in\Upsilon} \mathbf{\Psi}^{k,l}_{\mathbf{s}}(n_1,n_2)\mathbf{x}^T(n_1-k,n_2-l)\right]. \quad (20)$$

Considering the case when $\partial L/\partial \mathbf{B} = 0$, then postmultiplying both sides of this equation by $\mathbf{B}^T$, we obtain the set of equations

$$E_N\left[\sum_{(k,l)\in\Upsilon} \mathbf{\Psi}^{k,l}_{\mathbf{s}}(n_1,n_2).\mathbf{s}^T(n_1-k,n_2-l)\right] = \mathbf{I} \quad (21)$$

where $\mathbf{I}$ is the identity matrix.

It should be kept in mind that source separation is achieved, at best, up to a diagonal matrix. Therefore, the diagonal entries in (21) can be replaced by any scaling convention, and only the off-diagonal terms are important in (21). This finally yields the $K(K-1)$ following estimating equations:

$$E_N\left[\sum_{(k,l)\in\Upsilon} \psi^{k,l}_{s_i(n_1,n_2)}(n_1,n_2).s_j(n_1-k,n_2-l)\right] = 0 \quad i \neq j = 1,\ldots,K. \quad (22)$$

C. Equivariant Newton-Raphson Algorithm

In the classical version of the Newton-Raphson algorithm, one first assigns an initial value $\widetilde{\mathbf{B}}$ to the separating matrix $\mathbf{B}$. Then, the score functions in (22) are estimated around the source value $\tilde{\mathbf{s}} = \tilde{\mathbf{B}}\mathbf{x}$ using a first-order Taylor expansion. The aim is, therefore, to find a matrix $\mathbf{\Lambda}$ so that $\widetilde{\mathbf{B}} + \mathbf{\Lambda}$ is a solution of this first-order approximation.

It can be verified that the separation performance achieved with this algorithm depends on the mixing matrix, i.e., the more the K sources are mixed, the more separation is difficult. A particularly desirable property is, however, the equivariance of the separation algorithm [37], which means that the performance only depends on the initialization conditions and not on the actual mixing matrix. To derive an equivariant algorithm, we modify the updating rule $\hat{\mathbf{B}} = \tilde{\mathbf{B}} + \mathbf{\Lambda}$ of the Newton-Raphson algorithm into a serial update, so that the estimated separating matrix is proportional to its previous value $\widetilde{\mathbf{B}}$ according to the formula

$$\widehat{\mathbf{B}} = (\mathbf{I} + \mathbf{\Delta})\widetilde{\mathbf{B}} \quad (23)$$

where $\mathbf{\Delta}$ is some updating matrix defined below.

In the following, we will only consider the case $K = 2$ to simplify the notations. The extension of this approach to a higher number of source images is nevertheless straightforward.

Denoting $\mathbf{\Delta} = \begin{pmatrix} \delta_{11} & \delta_{12} \\ \delta_{21} & \delta_{22} \end{pmatrix}$, we prove that the off-diagonal terms δ_{12} and δ_{21} are solutions of the following set of linear equations, where $\tilde{\mathbf{s}} = \tilde{\mathbf{B}}\mathbf{x}$ (see Appendix B for a proof)

$$\begin{aligned}
&E_N\Bigg[\sum_{(k,l)\in\Upsilon} \psi^{(k,l)}_{s_1(n_1,n_2)}(\tilde{s}_1(n_1,n_2)|\tilde{s}_1(n_1,n_2-1),\ldots,\\
&\qquad \tilde{s}_1(n_1-1,n_2-1)).\tilde{s}_1(n_1-k,n_2-l)\Bigg]\delta_{21}\\
&\quad + E_N\Bigg[\sum_{(k,l)\in\Upsilon}\Bigg\{\sum_{(i,j)\in\Upsilon}\frac{\partial}{\partial s_1(n_1-i,n_2-j)}\psi^{(k,l)}_{s_1(n_1,n_2)}\\
&\qquad (\tilde{s}_1(n_1,n_2)|\tilde{s}_1(n_1,n_2-1),\ldots,\tilde{s}_1(n_1-1,n_2-1))\\
&\qquad \cdot \tilde{s}_2(n_1-i,n_2-j)\Bigg\}.\tilde{s}_2(n_1-k,n_2-l)\Bigg]\delta_{12}\\
&= -E_N\Bigg[\sum_{(k,l)\in\Upsilon}\psi^{(k,l)}_{s_1(n_1,n_2)}(\tilde{s}_1(n_1,n_2)|\tilde{s}_1(n_1,n_2-1),\ldots,\\
&\qquad \tilde{s}_1(n_1-1,n_2-1)).\tilde{s}_2(n_1-k,n_2-l)\Bigg] \qquad (24)\\
&E_N\Bigg[\sum_{(k,l)\in\Upsilon}\psi^{(k,l)}_{s_2(n_1,n_2)}(\tilde{s}_2(n_1,n_2)|\tilde{s}_2(n_1,n_2-1),\ldots,\\
&\qquad \tilde{s}_2(n_1-1,n_2-1)).\tilde{s}_2(n_1-k,n_2-l)\Bigg]\delta_{12}\\
&\quad + E_N\Bigg[\sum_{(k,l)\in\Upsilon}\Bigg\{\sum_{(i,j)\in\Upsilon}\frac{\partial}{\partial s_2(n_1-i,n_2-j)}\psi^{(k,l)}_{s_2(n_1,n_2)}\\
&\qquad (\tilde{s}_2(n_1,n_2)|\tilde{s}_2(n_1,n_2-1),\ldots,\tilde{s}_2(n_1-1,n_2-1))\\
&\qquad \cdot \tilde{s}_1(n_1-i,n_2-j)\Bigg\}.\tilde{s}_1(n_1-k,n_2-l)\Bigg]\delta_{21}\\
&= E_N\Bigg[\sum_{(k,l)\in\Upsilon}\psi^{(k,l)}_{s_2(n_1,n_2)}(\tilde{s}_2(n_1,n_2)|\tilde{s}_2(n_1,n_2-1),\ldots,\\
&\qquad \tilde{s}_2(n_1-1,n_2-1)).\tilde{s}_1(n_1-k,n_2-l)\Bigg]. \qquad (25)
\end{aligned}$$

It can be seen from the above set of equations that the off-diagonal entries δ_{12} and δ_{21} only depend on the vector $\widetilde{\mathbf{s}}$. The diagonal entries of the matrix $\mathbf{\Delta}$ can be set to any arbitrary value, for example zero, due to the scale indeterminacy of ICA. Thus, $\mathbf{\Delta}$ is only a function of $\widetilde{\mathbf{s}}$, which we denote $\mathbf{\Delta} = \phi(\widetilde{\mathbf{s}})$. Postmultiplying by $\mathbf{A}$ the updating rule in (23), then considering the global mixing-separating matrix $\hat{\mathbf{C}} = \hat{\mathbf{B}}\mathbf{A}$, we can rewrite (23) in the following form

$$\begin{aligned}
\widehat{\mathbf{C}} &= (\mathbf{I} + \phi(\widetilde{\mathbf{B}}x))\widetilde{\mathbf{B}}\mathbf{A}\\
&= (\mathbf{I} + \phi(\widetilde{\mathbf{B}}As))\widetilde{\mathbf{B}}\mathbf{A}\\
&= (\mathbf{I} + \phi(\widetilde{\mathbf{C}}s))\widetilde{\mathbf{C}} \qquad (26)
\end{aligned}$$

with $\tilde{\mathbf{C}} = \tilde{\mathbf{B}}\mathbf{A}$.

It can be easily verified that the global mixing-separating matrix, recursively obtained using the formula in (26), only depends on the initial value of the matrix $\tilde{\mathbf{C}}$, so that the algorithm satisfies the equivariance property.

Until now, we supposed that the score functions of the sources were known. Since the sources are not observable, their score functions can be estimated only via the reconstructed sources $\hat{\mathbf{s}}(n_1,n_2) = \widehat{\mathbf{B}}\mathbf{x}(n_1,n_2)$ at each step of the Newton-Raphson algorithm. This approximation is clearly not accurate at the first steps of the algorithm. However, our tests show that the algorithm is usually robust to this bad initialization and converges towards actual sources in a reasonable number of iterations.

In [18], a nonparametric spline kernel estimator, introduced in [38], was used. On the contrary, we adopt in the following section a parametric approach to estimate the score functions of the reconstructed sources and their derivatives. This estimator, based on third-order polynomial functions, is particularly interesting for its simplicity and computational efficiency,[6] and the estimation of the score function derivatives is straightforward in this case.

III. Parametric Estimation of the Score Functions

A. Score Function Polynomial Estimators

To simplify the notations, we will denote by $\psi^{k,l}_{\hat{s}_i(n)}(\xi_0|\xi_1,\ldots,\xi_4)$ the conditional score function at $\hat{s}_i(n_1,n_2)$, defined as in (15), and whose explicit notation reads

$$\begin{aligned}
&\psi^{k,l}_{\hat{s}_i(n_1,n_2)}(\hat{s}_i(n_1,n_2)|\hat{s}_i(n_1,n_2-1)\\
&\quad \hat{s}_i(n_1-1,n_2+1),\hat{s}_i(n_1-1,n_2)\\
&\quad \hat{s}_i(n_1-1,n_2-1)). \qquad (27)
\end{aligned}$$

Using rules for conditional densities, this conditional score function may be rewritten as

$$\begin{aligned}
&\psi^{k,l}_{\hat{s}_i(n)}(\xi_0|\xi_1,\ldots,\xi_4)\\
&\qquad = \psi^{k,l}_{\hat{s}_i(n)}(\xi_0,\ldots,\xi_4) - \psi^{k,l}_{\hat{s}_i(n)}(\xi_1,\ldots,\xi_4). \qquad (28)
\end{aligned}$$

Our objective is to find two parametric least mean square estimates, $g^{k,l}_{\hat{s}_i(n)}(\xi_0,\ldots,\xi_4,\mathbf{W})$ and $u^{k,l}_{\hat{s}_i(n)}(\xi_1,\ldots,\xi_4,\mathbf{V})$, of the joint score functions $\psi^{k,l}_{\hat{s}_i(n)}(\xi_0,\ldots,\xi_4)$ and $\psi^{k,l}_{\hat{s}_i(n)}(\xi_1,\ldots,\xi_4)$, respectively. This yields for the first estimator an optimization problem expressed as

$$\begin{aligned}
\mathbf{W}^{k,l}(\hat{s}_i(n)) = \arg\min_{\mathbf{W}} E\big\{&[\psi^{k,l}_{\hat{s}_i(n)}(\xi_0,\ldots,\xi_4)\\
&-g^{k,l}_{\hat{s}_i(n)}(\xi_0,\ldots,\xi_4,\mathbf{W})]^2\big\}. \qquad (29)
\end{aligned}$$

This criterion may be simplified using the following theorem, proved in [39] for univariate functions and extended to multivariate functions in [40].[7]

[6] On a 1.4 GHz AMD-Athlon PC with 512 MB of RAM, the polynomial estimation of the score function using 100×100 pixel images is more than 12 times faster than the estimation obtained with the spline kernel estimator. Moreover, the spline kernel estimator suffers from critically high memory consumption. For instance, using a PC with 1 GB of RAM, we do not have enough memory to compute the kernel spline estimate of a 200×200 pixel image, which is not the case for our polynomial estimator.

[7] The proof is based on integration by parts.

Theorem: For any arbitrary multivariate function $g(y_1,\ldots,y_q)$ that satisfies

$$\lim_{(y_1,\ldots,y_q)\longrightarrow\pm\infty} f_y(y_1,\ldots,y_q)g(y_1,\ldots,y_q)=0,$$

where $f_y(.)$ is the joint pdf of the variables $(y_1,\ldots,y_q)$, we have

$$E\left[-\frac{\partial \log f_y(y_1,\ldots,y_q)}{\partial y_i}g(y_1,\ldots,y_q)\right] = E\left[\frac{\partial g(y_1,\ldots,y_q)}{\partial y_i}\right] \quad \forall i=1,\ldots,q. \quad (30)$$

This theorem is actually satisfied by all real-world images. Thus, we can reformulate the optimization problem (29) as follows:

$$\mathbf{W}^{k,l}(\hat{s}_i(n)) = \arg\min_{\mathbf{W}}\left\{E\left[(g^{k,l}_{\hat{s}_i(n)}(\xi_0,\ldots,\xi_4,\mathbf{W}))^2\right] - 2E\left[\frac{\partial g^{k,l}_{\hat{s}_i(n)}(\xi_0,\ldots,\xi_4,\mathbf{W})}{\partial \xi^{k,l}}\right]\right\} \quad (31)$$

where $\xi^{k,l}\in\{\xi_0,\ldots,\xi_4\}$ and represents $\hat{s}_i(n_1-k,n_2-l)$ according to (15) and (27).[8] Note that the parametric estimator $g^{k,l}_{\hat{s}_i(n)}(.)$ may be chosen in different ways. However, polynomial functions seem to be advantageous in this case, due to their linearity with respect to the parameters in $\mathbf{W}$, the simplicity of their computation and derivation. Therefore, we here use third-order multivariate polynomial functions to estimate the joint score functions. The order of the polynomial function is chosen to provide a low computational cost without degrading estimation accuracy. The polynomial estimator $g^{k,l}_{\hat{s}_i(n)}(.)$ can be written as

$$g^{k,l}_{\hat{s}_i(n)}(\xi_0,\ldots,\xi_4,\mathbf{W}) = \sum_j w^{k,l}_j(\hat{s}_i(n))h_j(\xi_0,\ldots,\xi_4) = \mathbf{h}^T\mathbf{W}^{k,l}(\hat{s}_i(n)) \quad (32)$$

where $h_j(\xi_0,\ldots,\xi_4)$ and $w^{k,l}_j(\hat{s}_i(n))$ are respectively the monomial functions and the coefficients. Replacing $g^{k,l}_{\hat{s}_i(n)}(.)$ in (31) by its general polynomial formulation (32), we obtain

$$\mathbf{W}^{k,l}(\hat{s}_i(n)) = \arg\min_{\mathbf{W}}\left\{E\left[\mathbf{h}^T\mathbf{W}^{k,l}(\hat{s}_i(n))[\mathbf{W}^{k,l}(\hat{s}_i(n))]^T\mathbf{h}\right] - 2E\left[\frac{\partial \mathbf{h}^T}{\partial \xi^{k,l}}\mathbf{W}^{k,l}(\hat{s}_i(n))\right]\right\}. \quad (33)$$

It may be shown easily that, setting to zero the derivative of the above function with respect to the parameter vector $\mathbf{W}^{k,l}(\hat{s}_i(n))$, the minimum finally reads

$$\mathbf{W}^{k,l}_{LMS}(\hat{s}_i(n)) = \left(E\left[\mathbf{h}\mathbf{h}^T\right]\right)^{-1}E\left[\frac{\partial \mathbf{h}}{\partial \xi^{k,l}}\right]. \quad (34)$$

[8]This result shows a nice property of score functions: even though we have no knowledge about them, we can design a least mean square estimator for them.

Note that the parameters in (34) do not explicitly depend on the score functions $\psi^{k,l}$.

The estimate $u^{k,l}_{\hat{s}_i(n)}(\xi_0,\ldots,\xi_4,\mathbf{V})$ of the second joint score function, $\psi^{k,l}_{\hat{s}_i(n)}(\xi_1\ldots,\xi_4)$, can be computed in the same way. Thus, we define the polynomial estimate $u^{k,l}_{\hat{s}_i(n)}(.)$ by

$$u^{k,l}_{\hat{s}_i(n)}(\xi_1,\ldots,\xi_4,\mathbf{V}) = \sum_j v^{k,l}_j(\hat{s}_i(n))k_j(\xi_1,\ldots,\xi_4) = \mathbf{k}^T\mathbf{V}^{k,l}(\hat{s}_i(n)) \quad (35)$$

where $k_j(\xi_1,\ldots,\xi_4)$ and $v^{k,l}_j(\hat{s}_i(n))$ are respectively the monomial functions and the coefficients. Following the same steps as for the previous estimation, we finally obtain the following formula for the parameter vectors

$$\mathbf{V}^{k,l}_{LMS}(\hat{s}_i(n)) = \left(E\left[\mathbf{k}\mathbf{k}^T\right]\right)^{-1}E\left[\frac{\partial \mathbf{k}}{\partial \xi^{k,l}}\right] \quad (36)$$

where $\xi^{k,l}\in\{\xi_1,\ldots,\xi_4\}$.

The functions $\mathbf{h}(.)$ and $\mathbf{k}(.)$ required for (34) and (36) are third-order multivariate monomial functions, which respectively contain all possible terms in $\{1,(\xi_0,\ldots,\xi_4),(\xi_0,\ldots,\xi_4)^2,(\xi_0,\ldots,\xi_4)^3\}$ and $\{1,(\xi_1,\ldots,\xi_4),(\xi_1,\ldots,\xi_4)^2,(\xi_1,\ldots,\xi_4)^3\}$, where the exponents correspond to the orders of the monomials. Thus, the numbers of coefficients we must compute for each joint score function estimator are $\sum_{p=0}^{3}\binom{5+p-1}{p}=56$ and $\sum_{p=0}^{3}\binom{4+p-1}{p}=35$, respectively.[9]

Our experiments show that third-order polynomials insure fast and accurate estimation of the score functions. Indeed, while our estimator is faster than Pham's nonparametric spline kernel estimator [38], both approaches almost yield the same performance (see Appendix C). Therefore, using higher-order polynomials seem to be unnecessary, since it will considerably increase computational cost without probably yielding significant improvement in estimation performance. For instance, if we use fourth-order polynomials for estimating score functions, we will need 70 and 35 additional coefficients for estimating $\mathbf{h}(.)$ and $\mathbf{k}(.)$, respectively. On the contrary, first- and second-order polynomials are not enough accurate for estimating these polynomials as shown by our tests.

B. Estimation for Stationary Images

In a general context, the polynomial coefficients in $W^{k,l}$ and $V^{k,l}$ should be estimated at each site (n_1,n_2) of the image due to spatial score function variations. However, the expectations in (34) and (36) cannot be estimated in this case, since only one realization of the source images is available. Hence, some statistical hypotheses should be introduced to circumvent this difficulty.

In this section, we suppose that the images are stationary, i.e., their statistics remain constant all over the images. The pixel index (n_1,n_2) can then be omitted all along the previous analysis and, under ergodicity conditions, the expectations, required

[9]The numbers of coefficients $w^{k,l}$ and $v^{k,l}$ for a given power p, with $p\in\{0,\ldots,3\}$ correspond to the numbers of possible ways to choose p elements from 5 and 4 variables, respectively, when repetitions are allowed.

for computing (34) and (36), are simply replaced by a spatial average all over the image, yielding the following formula for $\mathbf{W}^{k,l}$:

$$\mathbf{W}_{LMS}^{k,l} = \left(\frac{1}{(N_1-1)(N_2-2)} \sum_{n_1=2}^{N_1} \sum_{n_2=2}^{N_2-1} [\mathbf{h}\mathbf{h}^T] \right)^{-1} \times \left(\frac{1}{(N_1-1)(N_2-2)} \sum_{n_1=2}^{N_1} \sum_{n_2=2}^{N_2-1} \left[\frac{\partial \mathbf{h}}{\partial \xi^{k,l}} \right] \right) \quad (37)$$

where $\mathbf{h}$ is the column vector containing the monomial functions $h_i(\hat{s}_i(n_1,n_2), \hat{s}_i(n_1,n_2-1), \hat{s}_i(n_1-1,n_2+1), \hat{s}_i(n_1-1,n_2), \hat{s}_i(n_1-1,n_2-1))$ defined in the previous section.

The polynomial coefficients in (36) are computed in the same manner and are constant all over the image.

C. More Realistic Models Adapted to Nonstationary Images

Even though practical, the stationary image assumption used above is far from being realistic. In fact, most of real-world images are clearly nonstationary and the models describing their statistics should, therefore, depend on the pixel position. Thus, to get closer to real conditions, we now propose extensions of the above method to nonstationary source images. To this end, we adapt two methods, respectively called blocking and kernel smoothing, which were used in [17] to handle nonstationarity for *temporally uncorrelated 1-D* sources.

1) Blocking Method: This approach essentially deals with slowly varying source statistics. In this case, the image can be split into $M_1 \times M_2$ sub-images I_j, so that each resulting sub-image may be supposed to be stationary. We can then replace the expectations in (34) and (36) by a local spatial average all over the sub-image, in the same way as in our above stationary Markovian approach. Therefore, the coefficients of the polynomial estimators of the score functions do not depend on the pixel position within each sub-image, and the score functions can be estimated once in each sub-image I_j.

2) Kernel Smoothing Method: In a kernel smoothing approach, the expectations required for (34) and (36) are computed in each pixel of the image by locally smoothing the pixel intensity with respect to its neighborhood. In the following, we denote for simplicity the expectations in (34), at each pixel $\hat{s}_i(n_1,n_2)$, by $E[\phi(\xi_0(n_1,n_2),\ldots,\xi_4(n_1,n_2))]$, where $\{\xi_0(n_1,n_2),\ldots,\xi_4(n_1,n_2)\}$ is the set of pixels $\{\hat{s}_i(n_1,n_2), \hat{s}_i(n_1,n_2-1), \hat{s}_i(n_1-1,n_2+1), \hat{s}_i(n_1-1,n_2), \hat{s}_i(n_1-1,n_2-1)\}$ and $\phi(.)$ is a general notation for the nonlinear functions contained by $\mathbf{h}$ in (34). The estimate of the above expectation is given by the following nonparametric estimator

$$\hat{E}[\phi(\xi_0(n_1,n_2),\ldots,\xi_4(n_1,n_2))] = \left(\sum_{\mu_1=2}^{N_1} \sum_{\mu_2=2}^{N_2-1} \kappa\left(\frac{\mu_1-n_1}{\nu}, \frac{\mu_2-n_2}{\nu} \right) \right)^{-1} \cdot \left[\sum_{\mu_1=2}^{N_1} \sum_{\mu_2=2}^{N_2-1} \kappa\left(\frac{\mu_1-n_1}{\nu}, \frac{\mu_2-n_2}{\nu} \right) \cdot \phi(\xi_0(\mu_1,\mu_2),\ldots,\xi_4(\mu_1,\mu_2)) \right] \quad (38)$$

where $\kappa(.)$ is a kernel function and ν is a window width parameter.

This estimator is very time consuming but should be more accurate than the blocking approach, since the estimates here are weighted averages around each image pixel, whereas the blocking method only uses the same sample mean over each sub-image. Nevertheless, if we modify our blocking algorithm, so that the supposedly stationary sub-images may be overlapping, both approaches turn out to be very similar. Indeed, using a rectangular kernel, the two methods lead to the same estimate. However, the kernel smoothing method offers more flexibility for estimation, since it allows us to modify pixel weights by changing the kernel distribution and/or the bandwidth parameter ν.

This estimator being very time consuming, we can reduce its computational cost by considering a sparser estimator defined by

$$\hat{E} = \left(\sum_{l_1=l_{11}}^{L_1} \sum_{l_2=l_{22}}^{L_2} \kappa\left(\frac{\frac{l_1 Q_1}{L_1} - n_1}{\nu}, \frac{\frac{l_2 Q_2}{L_2} - n_2}{\nu} \right) \right)^{-1} \cdot \left[\sum_{l_1=l_{11}}^{L_1} \sum_{l_2=l_{22}}^{L_2} \kappa\left(\frac{\frac{l_1 Q_1}{L_1} - n_1}{\nu}, \frac{\frac{l_2 Q_2}{L_2} - n_2}{\nu} \right) \cdot \phi\left(\xi_0\left(\frac{l_1 Q_1}{L_1}, \frac{l_2 Q_2}{L_2} \right), \ldots, \xi_4\left(\frac{l_1 Q_1}{L_1}, \frac{l_2 Q_2}{L_2} \right) \right) \right] \quad (39)$$

where $Q_1 = N_1$ and $Q_2 = N_2 - 1$. The sparseness parameters L_1 and L_2 are chosen so that Q_1/L_1 and Q_2/L_2 are integers, while l_{11} and l_{22} are the first integers such that $l_{11} \geq 2L_1/Q_1$ and $l_{22} \geq 2L_2/Q_2$.

The kernel smoothing estimator of (36) is obtained in the same way as above.

IV. Summary of the Algorithm

According to the previous analysis, the resulting BSS method operates as follows.

1) Initialize the separating matrix $\mathbf{B}$, for example using $\hat{\mathbf{B}} = \mathbf{I}$.
2) Repeat until $\hat{\mathbf{B}}$ does not change significantly, i.e until $\|\hat{\mathbf{B}}_{k+1} - \hat{\mathbf{B}}_k\|_\infty < \varepsilon$, where ε is a threshold parameter fixed in our tests to 10^{-5}:
   - estimate the reconstructed source images $\hat{\mathbf{s}} = \hat{\mathbf{B}}\mathbf{x}$;
   - center and normalize the output $\hat{\mathbf{s}}$, so that each of its components has unit variance;
   - estimate the conditional score functions in each pixel of the image using the polynomial estimators presented in Section III. Depending on the image, either stationary or nonstationary approaches are used to estimate the polynomial coefficients;
   - compute the derivatives of the conditional score functions, required for computing the coefficients of the set of (25);
   - solve the set of (25) to compute the off-diagonal entries of the matrix $\mathbf{\Delta}$, then update the estimated separating matrix according to $\widehat{\mathbf{B}}_{k+1} = (\mathbf{I} + \mathbf{\Delta})\widehat{\mathbf{B}}_k$.

V. Test Results

In the first two parts of this section, we present simulation results that we obtained with artificial images. In the next step, tests using artificial mixtures of real-world images are described and our method is compared to a set of classical approaches. An application to astrophysical image separation is finally depicted.

A. Combining Autocorrelation and Non-Gaussianity

In the following experiment, we compare our stationary bidimensional Markovian method to two classical BSS methods. The first one is the Pham-Garat algorithm [39], which uses an ML approach without taking into account the possible autocorrelation of the sources. Therefore, to perform separation, all sources except at most one must be non Gaussian. The second one is a 2-D extension of the SOBI algorithm [7], which is a second-order approach. This method, based on a joint diagonalization of several covariance matrices evaluated at different shifts, takes into account the autocorrelation of the sources, but not their possible non-Gaussianity.

We first generate two 100×100 pixel independent, white and uniformly distributed artificial bidimensional processes. These images, denoted $e_1(n_1, n_2)$ and $e_2(n_1, n_2)$, are then filtered by two autoregressive (AR) filters, according to the formula

$$\begin{aligned} s_i(n_1,n_2) = & e_i(n_1,n_2) + \rho^i_{0,1} s_i(n_1, n_2-1) \\ & + \rho^i_{1,-1} s_i(n_1-1, n_2+1) \\ & + \rho^i_{1,0} s_i(n_1-1, n_2) \\ & + \rho^i_{1,1} s_i(n_1-1, n_2-1). \end{aligned} \quad (40)$$

The coefficients of AR filters are chosen using a method proposed in [41] which guarantees their stability. Thus, the coefficients, $\rho^i_{j,k}$, are fixed to $\{-0.5, 0.4, 0.5, 0.3\}$ and $\{-0.5, \rho^2_{1,-1}, 0.5, 0.3\}$, respectively for $i = 1$ and 2. The coefficient $\rho^2_{1,-1}$ of the second filter may change in its stability interval, i.e., $[0.2, 0.6]$. Note that the generated source images follow exactly the fourth-order NSHP Markov model in this case.

The observations are finally obtained using the mixing matrix $\mathbf{A} = \begin{pmatrix} 1 & 0.99 \\ 0.99 & 1 \end{pmatrix}$. For each BSS algorithm tested here, the reconstructed sources $\hat{s}_i(n_1, n_2)$ are normalized by convention, so that they have the same variances and signs as the original source images. The separation performance is evaluated using the output Signal to Interference Ratio criterion (in dB) defined by

$$SIR = \frac{1}{K} \sum_{i=1}^{K} 10 \log_{10} \frac{E[s_i^2(n_1,n_2)]}{E[(s_i(n_1,n_2) - \hat{s}_i(n_1,n_2))^2]}. \quad (41)$$

In the present test, the number of sources is $K = 2$. The SIRs obtained by the three compared algorithms are computed for 100 Monte Carlo runs, where we randomly change the two processes $e_i(n_1, n_2)$. The mean SIR is shown in Fig. 3(a) as a function of the coefficient $\rho^2_{1,-1}$ of the second AR filter. Note that, whatever $\rho^2_{1,-1}$, our Markovian algorithm outperforms the other two. The performance of the Pham-Garat algorithm degrades while increasing the value of the filter coefficient $\rho^2_{1,-1}$, since the autocorrelation within the second source image thus becomes more important and this source approaches Gaussianity. Furthermore, we remark that the SOBI algorithm fails to separate the sources when we use the same filter for the two noise images, i.e $\rho^2_{1,-1} = 0.4$. This behavior of the algorithm is not surprising because the two sources have the same spectrum in this case.

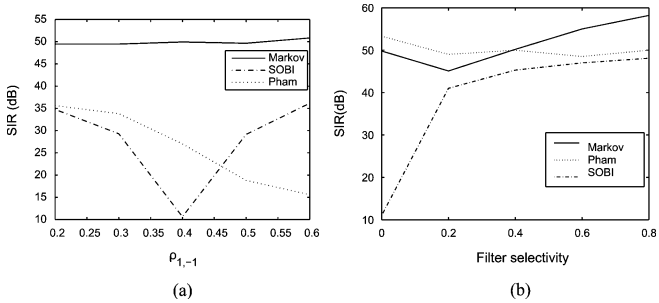

Fig. 3. Simulation results using (a) AR and (b) FIR filters.

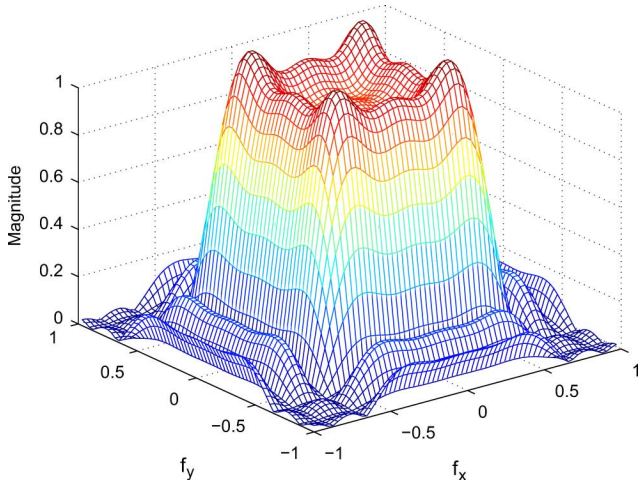

Fig. 4. FIR filter frequency response.

In the second part of this experiment, we generate similarly, two white, independent and uniformly distributed bidimensional processes, $e_1(n_1, n_2)$ and $e_2(n_1, n_2)$, but only filter one of them by a symmetrical Finite Impulse Response (FIR) filter to obtain the source images $s_1(n_1, n_2)$ and $s_2(n_1, n_2)$. The chosen FIR filter is a low-pass one with a tunable selectivity and an example of its response is shown in Fig. 4. The image filtering was practically done using a 9×9 pixel square mask. Note that in this case, the fourth-order NSHP MRF model is no longer satisfied by this source image. i.e., we here test our approach beyond the framework for which it was designed.

The resulting source images are then mixed by the same matrix $\mathbf{A}$ as above, and the mean SIR over 100 Monte Carlo runs is computed and plotted in Fig. 3(b) as a function of the selectivity of the FIR filter.

We remark that our algorithm always outperforms the SOBI algorithm. The Pham-Garat algorithm achieves slightly better results only when the selectivity is low, since the source images are nearly white in this case, so that their autocorrelation cannot be exploited by our Markovian algorithm. However, even in this

case, we still obtain good performance, with a 50-dB SIR, while SOBI provides poor performance, i.e., down to 10 dB.

From the results presented above, one can clearly see the benefit of using both non-Gaussianity and spatial autocorrelation when separating non-Gaussian autocorrelated images. Indeed, the fourth-order NSHP Markov random fields, that we used in our algorithm, provide a flexible spatial model, which can sufficiently describe the pixel interactions. The above last experiment especially proves the robustness of our method with respect to this neighborhood choice, since we obtain good separation performance even when the images do not exactly fit a fourth-order NSHP-MRF model.

However, this leaves an interesting question open: why do we limit our neighborhood to a fourth-order Markovian model? The relevance of this choice is proved by the following tests, where we compare the separation performance of our fourth-order Markovian method to different versions of the same approach modeling spatial autocorrelation with second- and sixth-order NSHP MRFs, respectively. The neighborhoods associated to these two models correspond to the two sets $V_1 = \{s_i(n_1 - 1, n_2), s_i(n_1, n_2 - 1)\}$ and $V_2 = \{s_i(n_1, n_2 - 1), s_i(n_1, n_2 - 2), s_i(n_1 - 1, n_2 + 1), s_i(n_1 - 1, n_2), s_i(n_1 - 1, n_2 - 1), s_i(n_1 - 2, n_2)\}$, respectively.[10]

The observations are generated in the same way as in the previous experiments, using two FIR filters of selectivity 0.4 and 0.8, according to Fig. 3. Over 100 Monte Carlo simulations, the fourth-order bidimensional Markovian method leads to a mean SIR of 48.5 dB, whereas we obtain 40.7 and 50.2 dB using the second- and sixth-order Markovian approaches, respectively. Note that the sixth-order Markovian method only yields a slight improvement, lower than 2 dB, with respect to the SIR obtained by the fourth-order method. On the contrary, the fourth-order method significantly outperforms the second-order approach, as it yields an improvement of about 8 dB of mean SIR. The computation times for the second-, fourth-, and sixth-order methods, on a 1.16 GHz AMD-Athlon PC with 512 MB of RAM, were typically equal to 12, 38, and 112 s, respectively, for each simulation. The time cost of the sixth-order method becomes especially critical when the image size increases. Indeed, switching from the above 100×100-sized images to 200×200 ones, the simulation times for the above three methods become 32, 129, and 15600 s, respectively. Hence, the choice of a fourth-order model is a good trade-off between computational cost and performance.

B. Improvement With Nonstationary Algorithm

In the second part of our experiments, we want to highlight the advantage of the nonstationary Markovian approach with respect to the stationary one when separating nonstationary images.

The tests are performed here using the blocking algorithm, presented in Section III. Following the same steps as in the previous simulations, we generate two 200×200 independent, white, and uniformly distributed artificial bidimensional processes, $e_1(n_1, n_2)$ and $e_2(n_1, n_2)$, that we filter using two AR filters according to the formula (40). The filter coefficients are chosen in the filter stability domains and are fixed to $\{-0.5, 0.3, 0.5, -0.29\}$ and $\{-0.5, 0.4, 0.5, 0.3\}$, respectively. The filtered images, denoted $\varsigma_1(n_1, n_2)$ and $\varsigma_2(n_1, n_2)$, are then split into 50×50 pixel sub-images, and the intensity of each such sub-image is multiplied by a different coefficient α_p^i, $p = 1, \ldots, 16$ to obtain the final sources $s_i(n_1, n_2)$. Note that the resulting source images are nonstationary and follow a fourth-order NSHP Markov model. The observations are finally obtained using the mixing matrix $\mathbf{A} = \begin{pmatrix} 1 & 0.99 \\ 0.99 & 1 \end{pmatrix}$.

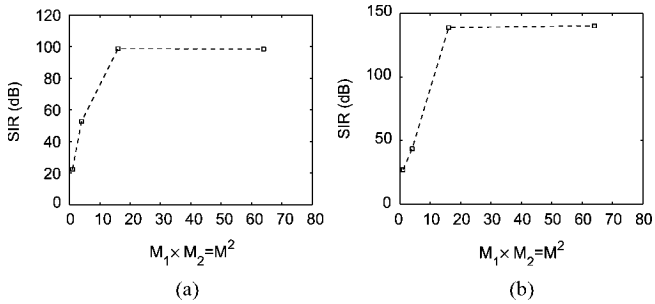

Fig. 5. Mean SIR over 100 Monte Carlo simulations versus number of sub-images for blocking algorithm applied to (a) AR-filtered source images and (b) FIR-filtered source images.

In the present experiment, the parameters M_1 and M_2 of the blocking algorithm are chosen to be equal and are denoted in the following by $M = M_1 = M_2$. Using our nonstationary blocking method for different values of the parameter M, we compute the average SIR over 100 Monte Carlo runs, where we change, for each run, the two random processes $e_i(n_1, n_2)$. The resulting SIRs are shown in Fig. 5(a) versus the number M^2 of sub-images.

The case $M = 1$ corresponds to our stationary Markovian method. We clearly see from Fig. 5(a) that neglecting the image nonstationarity considerably reduces separation performance. For example, the separation only yields a 22-dB mean SIR when using a one-block stationary algorithm, whereas it can reach a 99-dB average SIR when the number of blocks considered in the model is the same as in the actual images, i.e., $M_1 = M_2 = 4$. Furthermore, the separation performance is not degraded by over-blocking the image,[11] unless the number of pixels in each sub-image is too small to accurately estimate the score functions. For example, with $M_1 = M_2 = 20$, the separation fails and we obtain a mean SIR of only 3 dB.

We note that the nonstationary blocking method has the additional advantage of reducing memory requirements as compared to the basic stationary approach, since it operates with sub-images. For example, for the above simulation the stationary algorithm needs 192 Mbytes of memory when estimating the score functions, whereas the nonstationary version with $M = 4$ uses less than 12 Mbytes.

[10]The second- and sixth-order Markovian models are defined here according to the NSHP-MRF model definition, and so correspond to the number of pixels in the neighborhoods V_1 and V_2, respectively. Using the same convention as in noncausal MRF models, these two models are first- and third-order MRF models.

[11]The artificial image is said to be over-blocked when the number of blocks M^2 considered in the algorithm exceeds the actual number of blocks in this image.

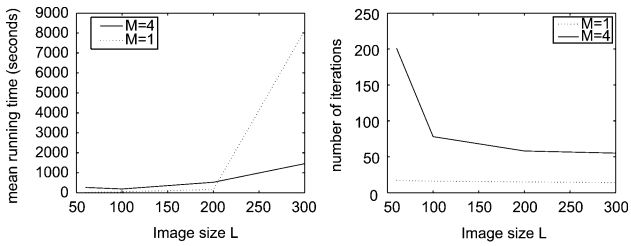

Fig. 6. Evolution of the overall complexity of the algorithms with respect to the size of images: (a) Mean of elapsed times per simulation for the stationary ($M = 1$) and the nonstationary blocking ($M = 4$) methods versus $L = \sqrt{N}$. (b) Average number of iterations to reach convergence for the stationary ($M = 1$) and the nonstationary blocking ($M = 4$) methods versus $L = \sqrt{N}$.

The computation time can also be reduced for large sized images by using the blocking method. For example, repeating the same experiment as above with 300×300-sized images, the overall running times, on a 1.16-GHz AMD-Athlon PC with 512 MB of RAM, are about 8200 s for the stationary Markovian algorithm and only 1500 s for the nonstationary version with $M = 4$. The mean of overall running time and the average number of iterations to reach convergence, for the above simulation with different image sizes, are computed over 100 simulations[12] and shown in Fig. 6(a) and 6(b), respectively, as a function of $L = \sqrt{N}$, where N is the number of pixels in each image.

Finally, we note that kernel smoothing algorithm is much more greedy than the blocking algorithm, even with small-sized images. For example, considering 50×50 images, the blocking nonstationary algorithm is more than 2000 times faster than the kernel smoothing approach. For this reason, only the nonstationary blocking algorithm is used in the following simulations.

In the second part of our experiment, we relax the Markovian model hypothesis in the simulated source images as in the previous section. Thus, two 200×200 independent, white, and uniformly distributed bidimensional processes are generated and then filtered by two different symmetrical bidimensional FIR filters. These filters are low-pass ones and are generated in the same way as in the previous experiments. The images are then split into 16 square sub-images and each such sub-image is multiplied by a different coefficient α_p^i. This finally yields two source images that are nonstationary, but cannot be perfectly modeled by NSHP Markov fields.

After mixing these sources by the same matrix $\mathbf{A}$ as in the previous test, the blocking algorithm is applied for different values of the number M^2 of sub-images and the mean SIR is computed over 100 Monte Carlo simulations and shown in Fig. 5(b) as a function of M^2. The algorithm leads to very high performance, with a 140-dB SIR, which proves the robustness of our approach with respect to the Markov model assumption. Note that the results obtained here surprisingly outperform those reported in the previous section. Nevertheless, this can be only due to the spectral diversity between the sources which is higher in this second experiment.

[12]The mean computation times and the average number of iterations were computed over 100 simulations, except for the stationary method with images of size 300×300, where we only used 6 simulations due to their high computational cost.

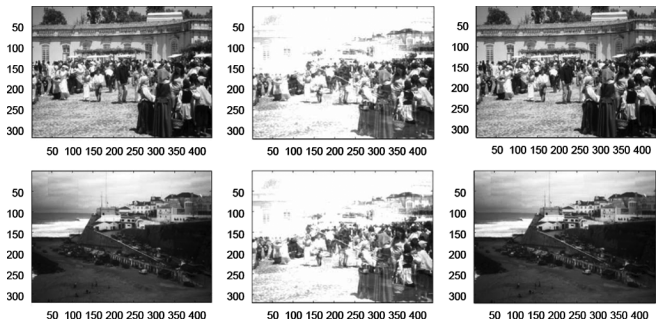

Fig. 7. Separating an artificial mixture of two real-world images: from left to right, original sources, observations resulting from an artificial linear instantaneous strong mixture, and reconstructed source images using our blocking method with $M^2 = 100$ sub-images.

The advantage of the nonstationary version is again highlighted in this case: only 27 dB are obtained as an average SIR when using a one-block Markovian algorithm, while we can reach a 140-dB mean SIR with a 16-block nonstationary algorithm, for example.

C. Separation of Artificial Mixtures of Real-World Images

In the previous experiments, the source images were generated artificially, so that they can fit the blocking nonstationarity condition. This hypothesis is, obviously, not satisfied when separating real-world images. Indeed, there is actually no blocking model which exactly fits the image variations. Nevertheless, since most real-life images are composed of adjacent homogeneous areas or objects, we can suppose the images to be almost locally stationary in these regions.

In the present simulation, we want to show that our blocking nonstationary approach is an acceptable first-order approximation of the image statistics which can model the nonstationarity in real-world images. To this end, we apply our blocking Markovian algorithm for separating an artificial mixture of two real-life images shown in Fig. 7. These images, available at [42], are 320×440-sized and clearly nonstationary. The mixing is artificially done with the matrix $\mathbf{A} = \begin{pmatrix} 1 & 0.99 \\ 0.99 & 1 \end{pmatrix}$. Using different values for the number of blocks M^2, the SIR obtained by our nonstationary blocking method is computed. For example, using a number of sub-images $M^2 = 100$, we obtain an SIR of nearly 70 dB, whereas the one-block version of the algorithm, which does not take into account nonstationarity, fails to separate these images. Moreover, the choice of M^2 is not very critical, since we still obtain more than 50-dB SIRs for a large range of numbers of sub-images.

This result is then compared to those achieved by the 15 algorithms available in the 2-D ICALAB Toolbox [2], [43]. Our blocking method highly outperforms all 15 algorithms, which only yield 37-dB SIR at best (with SOBI-RO algorithm). Furthermore, our algorithm clearly outperforms the Pham-Garat [39] and 2-D version of SOBI [7] algorithms, which only yield 13-dB and 25-dB SIR, respectively.

D. Astrophysical Component Separation

In the last experiment, we aim at separating astrophysical emissions using our Markovian methods.

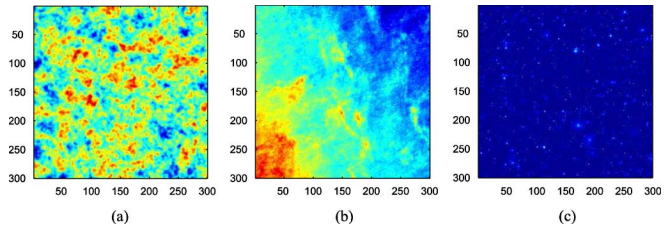

Fig. 8. Original spatial template of the astrophysical components: (a) CMB radiation, (b) dust emission, (c) Sunyaev-Zel'dovich (SZ) effect.

Since an initial work by Baccigialupi *et al.* [44], the contribution of BSS methods to separate superpositions of astrophysical components in sky maps has attracted the interest of the scientific community and many separation methods have been considered in the literature to deal with this crucial problem [45]–[49]. This was also motivated by the need for automatically processing large sets of observations which should be available in the future, thanks to space missions.

In particular, the Planck space mission, launched recently by the European Space Agency (ESA), will provide full sky maps in nine frequency channels in the range $[30 \cdots 850]$ GHz using a resolution of 5 to 30 arcminutes. One of the main objectives of this mission is to give an accurate description of the temperature anisotropies of the Cosmic Microwave Background (CMB) radiation, essentially important to constrain existing cosmological theories. Nevertheless, this requires data processing since the CMB radiation is usually superposed to other astrophysical components, called foregrounds. Moreover, these foregrounds have their own interest for astrophysical research, and so some efficient BSS techniques should be applied to obtain separately each of the mixed components.

In the present simulation, we limit our interest to the observations provided by the High Frequency Instrument (HFI) of the Planck satellite. Equipped with six detectors operating in six different frequency channels ranging from 100 to 850 GHz, the Planck HFI will simultaneously provide six observations of the sky map. In this frequency range, the observations mainly result from the superposition of only three astrophysical components, namely the CMB radiation, the dust emission and the Sunyaev-Zel'dovich (SZ) effect in clusters of galaxies [47].

To reduce computational cost, the observations are here limited to 300×300-pixel sky patches. The original sources are the same as in [47] and shown in Fig. 8. Following the mixture model in [44], the observations are obtained using a noiseless linear instantaneous mixture of the above three sources, which means that we neglect the noise contamination and the beam convolution in the instrument detectors. Even though idealistic, the results obtained with this simplified observation model are interesting, since they prove, as a first step, that the method is robust with respect to working hypotheses, i.e., Markovian and nonstationary image models, when applied to astrophysical images.

The mixing matrix coefficients, shown in Table I, correspond to those expected in the Planck HFI detectors and essentially depend on the source spectral emission and the channel frequencies [47]. The six resulting observations are shown in Fig. 9.

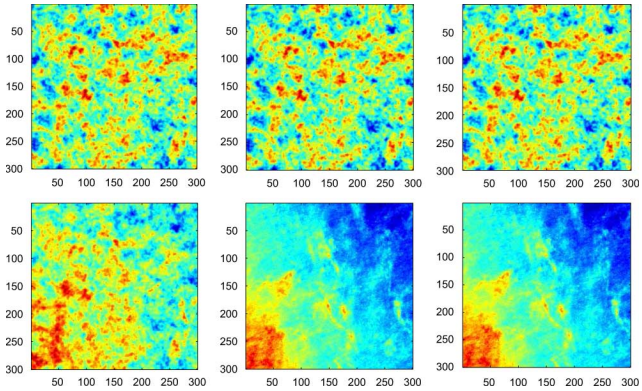

Fig. 9. Simulated observations from the six Planck HFI detectors with the expected mixing coefficients, in a noiseless context.

TABLE I
MIXING COEFFICIENTS USED IN OUR PLANCK HFI SIMULATION

Frequency (GHz)	100	143	217	353	545	857
CMB ($\times 10^{-6}$)	0.16	0.36	0.712	0.712	0.21	0.0084
Dust ($\times 10^{-6}$)	0.02	0.13	0.95	8.79	56.31	310.44
SZ ($\times 10^{-6}$)	-0.24	-0.377	0.0	1.59	1.18	0.0093

To perform separation using our Markovian method, we first reduce the number of observations to obtain the same as the number of sources. To this end, we apply Principal Component Analysis (PCA) [1] to the observations in Fig. 9, and then only keep the 3 highest-energy principal components.

The Markovian blocking method is then applied to the three resulting mutually uncorrelated images for different values of the number $M_1 \times M_2$ of sub-images. Whatever the number of blocks used in the algorithm, the Markovian method leads to high performance separation, with an average SIR higher than 43 dB, provided a sufficient number of samples in each block is available for score function estimation. The best performance is reached with $M_1 = 2$ and $M_2 = 3$, which yields a 55-dB average SIR. Nevertheless, the separation performance of the 3 astrophysical components does not evolve in the same way with respect to the variation of the number of blocks. First, the restoration of the CMB radiation is more accurate when using our stationary Markovian version, leading to nearly 59-dB SIR. Indeed, the separation performance for this component decreases a little when the image is split into blocks, due to the reduction of the number of samples used for score function estimation. On the contrary, the estimation of the dust emission is better when the nonstationary blocking Markovian algorithm is used, and the SIR improvement compared to that obtained by the stationary Markovian version is higher than 12 dB when using 100×100-pixel sub-images, for example. Note that this improvement is plausibly due to the nonstationarity of the dust component. Finally, the separation of the SZ emission does not necessarily improve when we use the nonstationary blocking Markovian algorithm, even though the SZ component is clearly nonstationary. This is not surprising because of the strong nonstationarity of the SZ component, which makes it a difficult task to split the image into stationary blocks with a sufficient number of samples. Using our kernel smoothing method may be actually interesting in this case, but the required running time is unfortunately too high. Thus, we here restrict our experiment to

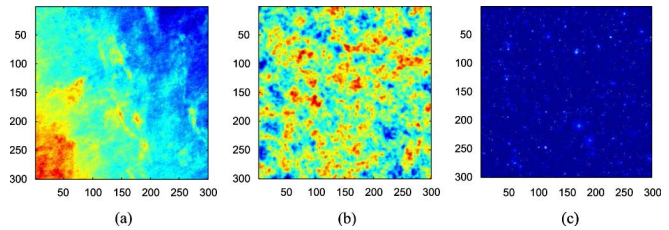

Fig. 10. Recovered source images: (a) dust emission, (b) CMB radiation, (c) SZ emission.

the nonstationary blocking Markovian method which proved, however, to be robust with respect to this strong nonstationarity, since it yields up to 58-dB SIR.

The three estimated components are shown in Fig. 10. Comparing them to Fig. 8 clearly proves that separation is quasi-perfect for all 3 components, since the difference between the original and estimated sources cannot be visually detected. The sources are not recovered in the original order, as a consequence of the permutation indeterminacy of BSS methods.

In the second step, we compare, as in the previous experiment, the results obtained with our method to those achieved by the 15 algorithms available in the 2-D ICALAB toolbox. The performance of the latter algorithms is quite poor, with a mean SIR lower than 13 dB for each method. The best performance among these algorithms is obtained by the SOBI-RO algorithm. This again proves the better performance of our method with respect to these classical algorithms.

As we mentioned before, the observed images were simulated in the above experiment according to an idealistic noiseless mixture model. Nonetheless, the actual observations on the Planck satellite will be contaminated by a significant level of instrumental noise. In the following, we test the robustness of our method with respect to additive Gaussian noise.

As in the previous simulation, three astrophysical components, namely CMB, dust emission and SZ effect, are mixed according to the coefficients expected on the 6 Planck-HFI frequency channels. Gaussian white noises $\epsilon_i, \forall i = 1, \ldots, 6$ are then added to these observations. The input signal-to-noise ratio (SNR) is defined, for each channel, as

$$SNR_i(dB) = 10log_{10}\frac{E[x_i^2(n_1, n_2)]}{E[\epsilon_i^2(n_1, n_2)]} \tag{42}$$

where x_i is the i-th noiseless observation and ϵ_i is the noise component on the i-th channel. Our aim is to study the separation performance of our blocking nonstationary Markovian method for different values of the mean input SNR over the 6 frequency channels, defined as

$$SNR = \frac{1}{6}\sum_{i=1}^{6} SNR_i \ . \tag{43}$$

First, we reduce the number of noisy observations by PCA, in the same way as in the previous experiment, in order to obtain the same number K for the observations and source images. The blocking Markovian method is then applied to the resulting images, for different values of the input SNR. This simulation is repeated for 100 random realizations of the noise components.

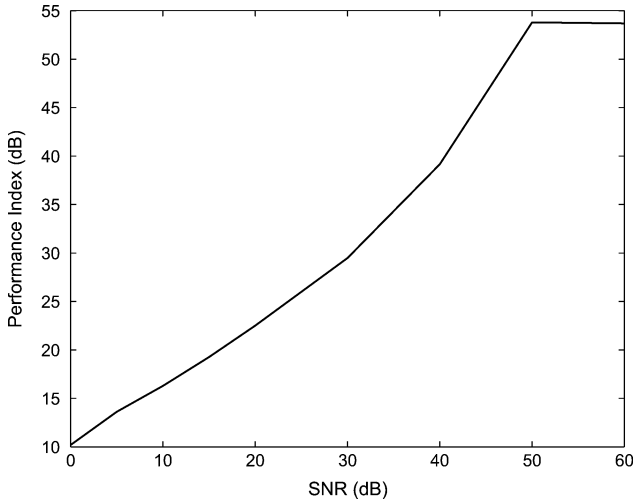

Fig. 11. Mean performance index obtained by our blocking nonstationary Markovian method as a function of the input SNR.

To evaluate separation performance, we compute the performance index for each source image, defined by

$$I_j(dB) = \max_i 10\log_{10}\left(\frac{c_{ij}^2}{\sum\limits_{k\neq j} c_{ik}^2}\right), \quad \forall j = 1, \ldots, K \tag{44}$$

where c_{ij} are the entries of the global separating-mixing matrix $C = BA$. The mean performance index for the three astrophysical components is then calculated as

$$I = \frac{1}{K}\sum_{j=1}^{K} I_j \tag{45}$$

and is shown in Fig. 11 as a function of the input SNR. It is worth noting that our method is, in theory, not adapted to this mixture model, since it is derived in a noiseless context. However, as we can see, results for moderate SNR are not so bad. For very low SNR, however, a noisy mixture model should be used to obtain good separation performance.

VI. Conclusion

In this paper, we presented a quasi-optimal blind image separation method based on an ML approach, wich deals with the linear instantaneous image mixture model. The autocorrelation within images has been described using a nonsymmetrical half-plane Markov random field model, which helped us to simplify the formulation of the conditional probability density function without introducing any prior information. The score functions were here computed using parametric LMS polynomial estimators. The estimating equations were then solved with an equivariant modified Newton-Raphson iterative algorithm. These choices essentially helped us to reduce the computational load and the memory consumption of our method.

The proposed Markovian method was developed in two steps. A simplified stationary approach was first presented, where the statistics of the source images were supposed to be invariant

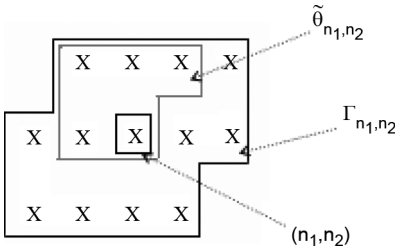

Fig. 12. Equivalence between a fourth-order NSHP Markov model and a noncausal MRF. $\Gamma_{n_1 n_2}$ is the noncausal neighborhood associated to the causal support set $\Theta_{n_1 n_2}$ defined in (10).

within the images. Extensions to more realistic nonstationary images were then proposed. Two different methods using either blocking or kernel smoothing were proposed to handle image nonstationarity. The improved nonstationary Markovian approach is particularly efficient, since it can simultaneously exploit the non-Gaussianity, autocorrelation and nonstationarity of the source images. Our experimental results, especially with artificially mixed real-world images, proved the high performance of our methods as compared to the classical algorithms available in the 2-D ICALAB toolbox.

Our kernel smoothing method is expected to yield high performance, but is very time consuming and some improvements are still needed to apply it to large-sized images.

The results achieved for the separation of a basic model of the Planck HFI observations were satisfactory. Therefore, we are now working on extending our Markovian nonstationary approach to the separation of noisy linear instantaneous mixtures. This new version will probably be implemented using the Expectation Maximization (EM) algorithm, and will be presented in future works.

Appendix A
Relationship Between a Fourth-Order NSHP Markov Model and Bidimensional MRFs

Defining for each site $(p,q) \in L$ the set $\tilde{\Theta}_{pq} = \Theta_{pq} \cup \{(p,q)\}$, and denoting by $\Lambda_{n_1 n_2} = \{(p,q)/(n_1,n_2) \in \tilde{\Theta}_{pq}\} = \{(n_1,n_2),(n_1,n_2+1),(n_1+1,n_2-1),(n_1+1,n_2),(n_1+1,n_2+1)\}$, it is proved in [29] that a fourth-order NSHP Markov model with the same support $\Theta_{n_1 n_2}$ as in (10) is equivalent to an MRF model with a noncausal neighborhood defined by

$$\Gamma_{n_1 n_2} = \{(i,j) \neq (n_1,n_2), (i,j) \in \tilde{\Theta}_{kl} \quad \forall (k,l) \in \Lambda_{n_1 n_2}\}$$

as shown in Fig. 12. This property of the NSHP Markov model is interesting, since it shows that the NSHP model indeed takes into account the autocorrelation between the pixel and all of its neighboring pixels.

Appendix B
Solving Estimating Equations

Starting from an initial value $\widetilde{\mathbf{B}}$ of the separating matrix, leading to an approximation $\widetilde{\mathbf{s}} = \widetilde{\mathbf{B}}\mathbf{x}$ of the actual sources, we aim at finding a matrix $\boldsymbol{\Delta}$ such that the new estimate of the separating matrix $\widehat{\mathbf{B}} = (\mathbf{I}+\boldsymbol{\Delta})\widetilde{\mathbf{B}}$ yields new source estimates

$$\widehat{\mathbf{s}} = \widehat{\mathbf{B}}\mathbf{x} = (\mathbf{I}+\boldsymbol{\Delta})\widetilde{\mathbf{s}} \tag{46}$$

which satisfy (22).

In the case $N = 2$, this yields the following two equations:

$$\begin{aligned}\widehat{s}_1(n_1,n_2) &= \widetilde{s}_1(n_1,n_2) + \delta_{11}\widetilde{s}_1(n_1,n_2) \\ &\quad + \delta_{12}\widetilde{s}_2(n_1,n_2) \\ \widehat{s}_2(n_1,n_2) &= \widetilde{s}_2(n_1,n_2) + \delta_{21}\widetilde{s}_1(n_1,n_2) \\ &\quad + \delta_{22}\widetilde{s}_2(n_1,n_2).\end{aligned} \tag{47}$$

To be independent, the estimated sources $\widehat{s}_1$ and $\widehat{s}_2$ should satisfy (22). Thus, using the above source updating rule to rewrite the first estimating equation at $\hat{s}_1(n_1,n_2)$, we obtain

$$\begin{aligned}E_N\Bigg[\sum_{(k,l)\in\Upsilon} &\{\psi^{k,l}_{s_1(n_1,n_2)}\big(\widetilde{s}_1(n_1,n_2) + \delta_{11}\widetilde{s}_1(n_1,n_2) \\ &+ \delta_{12}\widetilde{s}_2(n_1,n_2)|\widetilde{s}_1(n_1,n_2-1) + \delta_{11}\widetilde{s}_1(n_1,n_2-1) \\ &+ \delta_{12}\widetilde{s}_2(n_1,n_2-1),\ldots,\widetilde{s}_1(n_1-1,n_2-1) \\ &+ \delta_{11}\widetilde{s}_1(n_1-1,n_2-1) + \delta_{12}\widetilde{s}_2(n_1-1,n_2-1)\big)\big\} \\ &\cdot\big\{\widetilde{s}_2(n_1-k,n_2-l) + \delta_{21}\widetilde{s}_1(n_1-k,n_2-l) \\ &+ \delta_{22}\widetilde{s}_2(n_1-k,n_2-l)\big\}\Bigg] = 0.\end{aligned} \tag{48}$$

With a first-order Taylor expansion of the score function $\psi^{k,l}_{s_1(n_1,n_2)}(.)$, this equation becomes

$$\begin{aligned}E_N\Bigg[\sum_{(k,l)\in\Upsilon} &\Big\{\psi^{k,l}_{s_1(n_1,n_2)}\big(\widetilde{s}_1(n_1,n_2)|\widetilde{s}_1(n_1,n_2-1) \\ &\widetilde{s}_1(n_1-1,n_2+1),\widetilde{s}_1(n_1-1,n_2) \\ &\widetilde{s}_1(n_1-1,n_2-1)\big) + \sum_{(i,j)\in\Upsilon}\frac{\partial\psi^{k,l}_{s_1(n_1,n_2)}}{\partial s_1(n_1-i,n_2-j)} \\ &\big(\widetilde{s}_1(n_1,n_2)|\widetilde{s}_1(n_1,n_2-1),\widetilde{s}_1(n_1-1,n_2+1) \\ &\widetilde{s}_1(n_1-1,n_2),\widetilde{s}_1(n_1-1,n_2-1)\big) \\ &\cdot\big(\delta_{11}\widetilde{s}_1(n_1-i,n_2-j) + \delta_{12}\widetilde{s}_2(n_1-i,n_2-j)\big)\Big\} \\ &\cdot\Big\{\widetilde{s}_2(n_1-k,n_2-l) + \delta_{21}\widetilde{s}_1(n_1-k,n_2-l) \\ &+ \delta_{22}\widetilde{s}_2(n_1-k,n_2-l)\Big\}\Bigg] = 0.\end{aligned} \tag{49}$$

Neglecting second-order terms, this equation finally yields linear equations with respect to the entries of the matrix $\boldsymbol{\Delta}$

$$(1+\delta_{22})\mathbf{J}_1 + \delta_{21}\mathbf{J}_2 + \delta_{11}\mathbf{J}_3 + \delta_{12}\mathbf{J}_4 = 0 \tag{50}$$

where

$$\mathbf{J}_1 = E_N\Bigg[\sum_{(k,l)\in\Upsilon}\psi^{k,l}_{s_1(n_1,n_2)}\big(\widetilde{s}_1(n_1,n_2)|\widetilde{s}_1(n_1,n_2-1)$$

$$\ldots,\widetilde{s}_1(n_1-1,n_2-1)\big)\widetilde{s}_2(n_1-k,n_2-l)\Big]$$

$$\mathbf{J}_2 = E_N\Bigg[\sum_{(k,l)\in\Upsilon}\psi^{k,l}_{s_1(n_1,n_2)}\big(\widetilde{s}_1(n_1,n_2)|\widetilde{s}_1(n_1,n_2-1)\ldots,\widetilde{s}_1(n_1-1,n_2-1)\big)\widetilde{s}_1(n_1-k,n_2-l)\Bigg] \quad (51)$$

$$\mathbf{J}_3 = E_N\Bigg[\sum_{(k,l)\in\Upsilon}\Bigg[\sum_{(i,j)\in\Upsilon}\frac{\partial\psi^{k,l}_{s_1(n_1,n_2)}}{\partial s_1(n_1-i,n_2-j)}\big(\widetilde{s}_1(n_1,n_2)|\widetilde{s}_1(n_1,n_2-1),\ldots,\widetilde{s}_1(n_1-1,n_2-1)\big)\widetilde{s}_1(n_1-i,n_2-j)\Bigg]\widetilde{s}_2(n_1-k,n_2-l)\Bigg]$$

$$\mathbf{J}_4 = E_N\Bigg[\sum_{(k,l)\in\Upsilon}\Bigg[\sum_{(i,j)\in\Upsilon}\frac{\partial\psi^{k,l}_{s_1(n_1,n_2)}}{\partial s_1(n_1-i,n_2-j)}\big(\widetilde{s}_1(n_1,n_2)|\widetilde{s}_1(n_1,n_2-1),\ldots,\widetilde{s}_1(n_1-1,n_2-1)\big)\widetilde{s}_2(n_1-i,n_2-j)\Bigg]\widetilde{s}_2(n_1-k,n_2-l)\Bigg] \quad (52)$$

δ_{22} may be neglected with respect to 1 in (50). Furthermore, the estimated sources are centered and nearly independent in the vicinity of the solution, so that for any function Φ

$$E_N\left[\Phi(\widetilde{s}_1(n_1-k,n_2-l))\cdot\widetilde{s}_2(n_1-i,n_2-j)\right] \simeq E_N\left[\Phi(\widetilde{s}_1(n_1-k,n_2-l))\right]\cdot E_N\left[\widetilde{s}_2(n_1-i,n_2-j)\right]$$

is small.

It follows that $\delta_{11}\mathbf{J}_3$ is negligible with respect to the other terms in (50). Taking into account all these simplifications, we obtain the first linear equation defined in (25). The second equation is then derived by symmetry.

For more than two source images, the estimating equations are obtained following the same steps as above. Using a first-order Taylor expansion of the score functions at the current source values, and after some simplifications, we obtain $K(K-1)$ linear equations of the entries of $\mathbf{\Delta}$ defined in (46), which are denoted δ_{ij}, $\forall i,j=1,\ldots,K$ [see (53), shown at the bottom of the page]. These equations may be simplified by considering the sources to be nearly independent at the vicinity of a solution. Since the sources are centered, for any function $\phi(.)$, terms of the form $E_N[\phi(s_i(.))s_m(.)]\delta_{jm}$, $\forall m\neq i$ or $E_N[\phi(s_i(.))s_j(.)s_m(.)]\delta_{im}$, $\forall m\neq j$ are quite small, so that they can be neglected in (53).

Using these simplifications, we finally obtain, for each couple of source images s_i and s_j, two linear equations with two unknowns δ_{ij} and δ_{ji}, defined exactly in the same manner as in (25).

APPENDIX C
COMPARISON BETWEEN SCORE FUNCTION ESTIMATORS

In the following, we compare the score function estimate given by our third-order multivariate polynomial estimator to that obtained by Pham's nonparametric estimator[38]. The multivariate conditional score function, $\psi(\xi_0|\xi_1,\xi_2,\xi_3,\xi_4)$, for a 100×100-pixel real image, is estimated using both approaches, and then displayed as a function of the first variable ξ_0 in Fig. 13.

Fig. 13 shows that the results obtained with both estimators are quite similar, which confirms the relevance of our estimator. Indeed, the relative error between both estimates is about 6%.

$$\begin{aligned}
&\sum_{m=1}^{K}\delta_{jm}E_N\Bigg[\sum_{(k,l)\in\Upsilon}\psi^{k,l}_{s_i(n_1,n_2)}(\tilde{s}_i(n_1,n_2)|\tilde{s}_i(n_1,n_2-1)\\
&\quad\tilde{s}_i(n_1-1,n_2+1),\tilde{s}_i(n_1-1,n_2)\\
&\quad\tilde{s}_i(n_1-1,n_2-1))\tilde{s}_m(n_1-k,n_2-l)\Bigg]\\
&+\sum_{m=1}^{K}\delta_{im}E_N\Bigg[\sum_{(k,l)\in\Upsilon}\sum_{(\alpha,\beta)\in\Upsilon}\frac{\partial\psi^{k,l}_{s_i(n_1,n_2)}}{\partial s_i(n_1-\alpha,n_2-\beta)}\big(\widetilde{s}_i(n_1,n_2)|\\
&\quad\widetilde{s}_i(n_1,n_2-1),\widetilde{s}_i(n_1-1,n_2+1),\widetilde{s}_i(n_1-1,n_2),\widetilde{s}_i(n_1-1,n_2-1)\big)\\
&\quad\tilde{s}_m(n_1-\alpha,n_2-\beta).\tilde{s}_j(n_1-k,n_2-l)\Bigg]\\
&\quad=-E_N\Bigg[\sum_{(k,l)\in\Upsilon}\psi^{k,l}_{s_i(n_1,n_2)}(\tilde{s}_i(n_1,n_2)|\tilde{s}_i(n_1,n_2-1)\\
&\quad\tilde{s}_i(n_1-1,n_2+1),\tilde{s}_i(n_1-1,n_2)\\
&\quad\tilde{s}_i(n_1-1,n_2-1)).\tilde{s}_j(n_1-k,n_2-l)\Bigg]\\
&\quad\forall i\neq j=1,\ldots,K
\end{aligned} \quad (53)$$

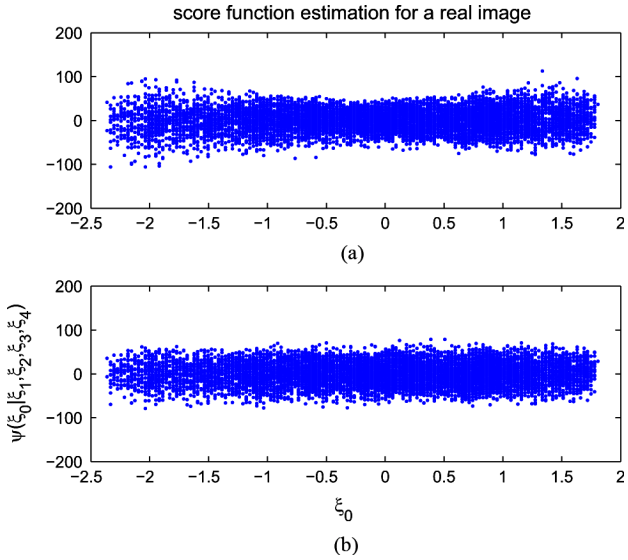

Fig. 13. Estimates of the conditional score function $\psi(\xi_0|\xi_1,\xi_2,\xi_3,\xi_4)$ versus ξ_0 for a 100 × 100-pixel real image. (a) Polynomial. (b) Pham.

ACKNOWLEDGMENT

The authors would like to thank Dr. H. Snoussi for providing the expected Planck HFI mixture coefficients and the generated spatial templates of the CMB radiation, dust emission, and Sunyaev-Zel'dovich (SZ) effect in clusters of galaxies [50].

REFERENCES

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

[2] A. Cichocki and S. Amari, *Adaptive Blind Signal and Image Processing: Learning Algorithms and Applications*. New York: Wiley, 2003.

[3] L. Bedini, D. Herranz, E. Salerno, C. Baccigulapi, E.-E. Kuruoglu, and A. Tonazzini, "Separation of correlated astrophysical sources using multiple-lag data covariance matrices," *EURASIP J. Appl. Signal Process.*, vol. 2005, no. 15, pp. 2400–2412, 2005.

[4] J.-F. Cardoso, "The three easy routes to independent component analysis: Contrast and geometry," in *Proc. Int.Conf. Independent Component Analysis and Blind Source Separation ICA*, San Diego, CA, Dec. 2001, pp. 1–6.

[5] L. Tong, R.-W. Liu, V. Soon, and Y.-F. Huang, "Indeterminacy and identifiability of blind identification," *IEEE Trans. Circuits Syst.*, vol. 38, no. 7, pp. 499–509, Jul. 1991.

[6] L. Molgedey and H. G. Shuster, "Separation of a mixture of independent signals using time delayed correlation," *Phys. Rev. Lett.*, vol. 72, no. 23, pp. 3634–3636, 1994.

[7] A. Belouchrani, K. A. Meraim, J.-F. cardoso, and E. Moulines, "A blind source separation technique based on second order statistics," *IEEE Trans. Signal Process.*, vol. 45, no. 2, pp. 434–444, Feb. 1997.

[8] A. Ziehe and K. Müller, "TDSEP—an efficient algorithm for blind separation using time structure," in *Proc. Int. conf. Artificial Neural Networks*, Skövde, Sweden, Dec. 1998, pp. 675–680.

[9] D.-T. Pham, "Blind separation of instantaneous mixture of sources via the Gaussian mutual information criterion," *Signal Process.*, vol. 81, no. 4, pp. 855–870, 2001.

[10] S.-I. Amari, "Estimating functions of independent component analysis for temporally correlated signals," *Neural Comput.*, vol. 12, no. 9, pp. 2083–2107, 2000.

[11] K. Matsuoka, M. Ohya, and M. Kawamoto, "A neural net for blind separation of non-stationary signals," *Neural Netw.*, vol. 8, no. 3, pp. 411–419, 1995.

[12] S. Choi, A. Cichocki, and S. Amari, "Equivariant non-stationary source separation," *Neural Netw.*, vol. 15, no. 1, pp. 121–130, 2002.

[13] A. Souloumiac, "Blind source detection and separation using second-order non-stationarity," in *Proc. IEEE Int. Conf. Acoustics, Speech, and Signal Processing*, Detroit, MI, May 1995, pp. 1912–1915.

[14] S. Choi, A. Cichoki, and A. Belouchrani, "Second order non-stationary source separation," *J. VLSI Signal Process.*, vol. 32, no. 1-2, pp. 93–104, 2002.

[15] A. Hyvarinen, "Blind source separation by non-stationarity of variance: A cumulant based approach," *IEEE Trans. Neural Netw.*, vol. 12, no. 6, pp. 1471–1474, Dec. 2001.

[16] D.-T. Pham and J.-F. Cardoso, "Blind separation of instantaneous mixtures of non-stationary sources," *IEEE Trans. Signal Process.*, vol. 49, no. 9, pp. 1837–1848, Sep. 2001.

[17] D.-T. Pham, "Blind separation of non stationary non Gaussian sources," in *Proc. 11th Eur. Signal Processing Conf.*, Toulouse, France, Sep. 2002, pp. 67–70.

[18] S. Hosseini, C. Jutten, and D.-T. Pham, "Markovian source separation," *IEEE. Trans. Signal Process.*, vol. 51, no. 12, pp. 3009–3019, Dec. 2003.

[19] S. Geman and D. Geman, "Stochastic relaxation, Gibbs distributions and the Bayesian restoration of images," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 6, no. 6, pp. 721–741, Dec. 1984.

[20] A. Tonnazini, L. Bedini, and E. Salerno, "A Markov model for blind image separation by mean-field EM algorithm," *IEEE Trans. Image Process.*, vol. 15, pp. 473–482, 2006.

[21] N. Bali and A. Mohammad-Djafari, "Bayesian approach with hidden Markov modeling and mean field approximation for hyperspectral data analysis," *IEEE Trans. Image Process.*, vol. 17, pp. 217–225, 2008.

[22] S. Jia and Y. Qian, "An MRF-ICA based algorithm for image separation," in *Proc. Advances in Natural Computation: 1st Int. Conf.*, Changsha, China, Aug. 2005, pp. 391–395.

[23] P. Brémaud, *Markov Chains: Gibbs Fields, Monte Carlo Simulation, and Queues*. New York: Springer-Verlag, 1999.

[24] K. Abend, T.-J. Harley, and L.-N. Kanal, "Classification of binary random patterns," *IEEE. Trans. Inf. Theory*, vol. 11, no. 4, pp. 538–544, 1965.

[25] L.-N. Kanal, "Markov mesh models," *Comput. Graph. Image Process.*, vol. 12, pp. 371–375, 1980.

[26] H. Derin, H. Elliott, R. Cristi, and D. Geman, "Bayes smoothing algorithms for segmentation of binary images modeled by Markov random fields," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. PAMI-6, no. 6, pp. 707–720, Jun. 1984.

[27] D. Preuss, "Two-dimensional Facsimile source encoding based on a Markov model," *Nachrichten Technische Zeitschrift (NTZ)*, vol. 28, no. 10, pp. 358–363, 1975.

[28] R. Chellapa and R.-L. Kashyap, "Digital image restoration using spatial interaction models," *IEEE. Trans. Acoust., Speech, Signal Process.*, vol. ASSP-30, no. 3, pp. 461–472, Mar. 1982.

[29] H. Derin and P.-A. Kelly, "Discrete-index Markov-type random processes," *Proc. IEEE*, vol. 77, pp. 1485–1510, 1989.

[30] F.-C. Jeng and J.-W. Woods, "On the relationship of the Markov mesh to the NSHP Markov chain," *Pattern Recognit. Lett.*, vol. 5, no. 4, pp. 273–279, 1987.

[31] S. Hosseini, R. Guidara, Y. Deville, and C. Jutten, "Markovian blind image separation," in *Proc. ICA*, Charleston, SC, Mar. 2006, pp. 106–114.

[32] S. Hosseini, R. Guidara, Y. Deville, and C. Jutten, "Maximum likelihood separation of spatially autocorrelated images using a Markov model," in *Proc. 26th Int. Workshop on Bayesian Inference and Maximum Entropy Methods in Science and Engineering (MaxEnt 2006)*, Paris, France, Jul. 2006, pp. 105–112.

[33] R. Guidara, S. Hosseini, and Y. Deville, "Markovian blind separation of non-stationary temporally correlated sources," in *Proc. 15th Eur. Symp. Artificial Neural Networks*, Bruges, Belgium, Apr. 2007, pp. 241–246.

[34] R. Guidara, S. Hosseini, and Y. Deville, "Blind separation of non-stationary images using Markov models," in *Proc. ICA*, London, U.K., Sep. 2007, pp. 722–729.

[35] R. Guidara, S. Hosseini, and Y. Deville, "Séparation aveugle de sources Markoviennes et non-stationnaires," in *Proc.* $21^{éme}$ *Colloq. GRETSI*, Troyes, France, Sep. 2007, pp. 1001–1004.

[36] C.-F. Caiafa, E. Salerno, A.-N. Proto, and L. Fiumi, "Blind spectral unmixing by local maximization of non-gaussianity," *Signal Process.*, vol. 88, no. 1, pp. 50–68, 2008.

[37] J.-F. Cardoso and B. Laheld, "Equivariant adaptive source separation," *IEEE Trans. Signal Process.*, vol. 44, no. 12, pp. 3017–3030, Dec. 1996.
[38] D.-T. Pham, "Fast algorithm for estimating mutual information, entropies and score functions," in *Proc. ICA*, Nara, Japan, Apr. 2003, pp. 17–22.
[39] D.-T. Pham and P. Garat, "Blind separation of mixture of independent sources through a quasi-maximum likelihood approach," *IEEE Trans. Signal Process.*, vol. 45, no. 7, pp. 1712–1725, Jul. 1997.
[40] M. Babaie-Zadeh, "On blind source separation in convolutive and nonlinear mixtures," Ph.D. dissertation, LIS-INPG, Grenoble, France, 2002.
[41] M. Benidir and M. Barret, *Stabilité des Filtres et des Systèmes Linéaires, (in French)*. Paris, France: Dunod, 1999, pp. 213–235.
[42] Web site of IcaCentral [Online]. Available: http://www.tsi.enst.fr/icacentral/base_multi.html
[43] A. Cichocki *et al.*, Icalab Toolboxes [Online]. Available: http://www.bsp.brain.riken.jp/ICALAB
[44] C. Baccigalupi *et al.*, "Neural networks and the separation of cosmic microwave background and astrophysical signals in sky maps," *Monthly Notices Roy. Astron. Soc.*, vol. 318, no. 7, pp. 769–780, 2000.
[45] D. Maino *et al.*, "All-sky astrophysical component separation with fast independent component analysis (fastica)," *Monthly Notices Roy. Astron. Soc.*, vol. 334, no. 1, pp. 53–68, 2002.
[46] J. Delabrouille, J.-F. Cardoso, and G. Patanchon, "Multi-detector multi-component spectral matching and applications for CMB data analysis," *Monthly Notices Roy. Astron. Soc.*, vol. 346, no. 4, pp. 1089–1102, 2003.
[47] H. Snoussi, G. Patanchon, J.-M. Pérez, A. Mohammad-Djafari, and J. Delabrouille, "Bayesian blind component separation for cosmic microwave background observations," in *Proc. MAXENT Workshops*, Baltimore, MD, Aug. 2001, pp. 125–140.
[48] Y. Moudden, J.-F. Cardoso, J.-L. Starck, and J. Delabrouille, "Blind component separation in wavelet space. Application to CMB analysis," *EURASIP J. Appl. Signal Process.*, vol. 2005, no. 15, pp. 2437–2454, 2005.
[49] H. Snoussi, "Fast MCMC spectral matching separation in noisy gaussian mixtures. Application to astrophysics," presented at the IEEE Int. Symp. Control, Communications, and Signal Processing, Marrakech, Morocco, Mar. 2006.
[50] H. Snoussi, "Approche Bayésienne en Séparation de Sources. Applications en Imagerie" Ph.D. dissertation, Univ. Paris Sud, Orsay, Paris, France, 2003.

Rima Guidara was born in Sfax, Tunisia, in 1982. In 2005, she graduated as an electronic engineer from the Ecole Nationale Supérieure d'Electrotechnique, d'Electronique, d'Informatique, d'Hydraulique et des Télécommunications, Toulouse, France, and received the M.Sc. degree in image and signal processing from the Institut National Polytechnique de Toulouse and the Ph.D. degree in signal and image processing from the University of Toulouse in 2009.

From 2005 to 2009, she was with the Laboratory of Astrophysics Toulouse-Tarbes, which is part of the French National Center for Scientific Research (CNRS) and also part of the Midi-Pyrénées Observatory and the University Paul Sabatier Toulouse 3. Her main field of research is blind source separation of signal and images.

Shahram Hosseini was born in Shiraz, Iran, in 1968. He received the B.Sc. and M.Sc. degrees in electrical engineering from the Sharif University of Technology, Tehran, Iran, in 1991 and 1993, respectively, and the Ph.D. degree in signal processing from the Institut National Polytechnique, Grenoble, France, in 2000.

He is currently an Associate Professor at the Université Paul Sabatier Toulouse 3, Toulouse, France. His research interests include BSS, artificial neural networks, and adaptive signal processing.

Yannick Deville (M'90) was born in Lyon, France, in 1964. He graduated from the Ecole Nationale Supérieure de Télécommunications de Bretagne, Brest, France, in 1986. He received the D.E.A. and Ph.D. degrees, both in microelectronics, from the University of Grenoble, Grenoble, France, in 1986 and 1989, respectively.

From 1986 to 1997, he was a Research Scientist at Philips Research Labs, Limeil, France. His investigations during this period concerned various fields, including GaAs integrated microwave RC active filters, VLSI cache memory architectures and replacement algorithms, neural network algorithms and applications, and nonlinear systems. Since 1997, he has been a Professor at the Université Paul Sabatier Toulouse 3, Toulouse, France. From 1997 to 2004, he was with the Acoustics Lab, Université Paul Sabatier Toulouse 3. Since 2004, he has been with the Astrophysics Lab in Toulouse, which is a part of the University but also of the French National Center for Scientific Research (CNRS) and of the Midi-Pyrénées Observatory. His current major research interests include signal processing, higher order statistics, time-frequency analysis, neural networks, and, particularly, BSS and ICA methods and their applications to astrophysics, acoustics, and communication/electromagnetic signals.