Inlier-Based ICA with an Application to Superimposed Images

Frank C. Meinecke,[1] Stefan Harmeling,[1] Klaus-Robert Müller[1,2]

[1] Fraunhofer FIRST.IDA, Kekuléstr. 7, Berlin, Germany

[2] Department of Computer Science, University of Potsdam, Potsdam, Germany

Received 8 June 2004; accepted 27 January 2005

ABSTRACT: This paper proposes a new independent component analysis (ICA) method which is able to unmix overcomplete mixtures of sparce or structured signals like speech, music or images. Furthermore, the method is designed to be robust against outliers, which is a favorable feature for ICA algorithms since most of them are extremely sensitive to outliers. Our approach is based on a simple outlier index. However, instead of robustifying an existing algorithm by some outlier rejection technique we show how this index can be used directly to solve the ICA problem for super-Gaussian sources. The resulting inlier-based ICA (IBICA) is outlier-robust by construction and can be used for standard ICA as well as for overcomplete ICA (i.e. more source signals than observed signals). Int J Imaging Syst Technol, 15, 48–55, 2005; Published online in Wiley InterScience (www.interscience.wiley.com). DOI 10.1002/ima.20037

Key words: Independent Component Analysis (ICA); Blind Source Separation (BSS); overcomplete ICA; outlier robustness

I. INTRODUCTION

Independent component analysis (ICA) is a blind signal separation technique that can be applied to many application domains. In image processing, ICA provided an automated way to obtain a suitable basis for sparse image coding, which resembles Gabor functions (e.g. Olshausen and Field, 1996 and also Hyvarinen et al., 2001). This paper uses ICA directly to separate images that might be corrupted by outliers or noise.

ICA models some measured multi-variate signals $x_n(t)$ with $n = 1, \ldots, N$ as a linear combination of statistically independent source signals $s_m(t)$ with $m = 1, \ldots, M$:

$$x_n(t) = \sum_m A_{nm} s_m(t). \qquad (1)$$

These signals can be time series, images or movies. The index t can be a time index or an index specifying a pixel in a digital image. The task of an ICA algorithm is to estimate the *mixing matrix* A given only the observations $x(t)$. Typically, it is assumed that $M \leq N$ and that the columns of A are linearly independent. In this case, Eq. (1) is invertible and the source signals $s(t)$ can be recovered.*

In the overcomplete† case, where the number of sources exceeds the number of mixtures (i.e. $M > N$), it is often still possible to identify the mixing matrix A, as long as the sources are super-Gaussian or sparse. However, in general the source signals cannot be recovered, since the model Eq. (1) is not invertible. For very sparse signals (or signals that can be represented sparsely, Lee et al., 1999; Bofill and Zibulevsky, 2001; Zibulevsky and Pearlmutter, 2001) the underdetermined blind source separation problem is solvable, because each data point can be uniquely assigned to one source (at least approximately). The strong spatial structure that is apparent in nearly all images makes it often possible to sparsify them by a suitable transformation. Standard sparsifying approaches are the use of Fourier transformations, spectrogram methods or wavelets (the latter we used in the experiment section, see Chen et al., 1998; Lee et al., 1999; Zibulevsky and Pearlmutter, 2001).

ICA algorithms typically make use of statistical properties of the projections (i.e. kurtosis, negentropy, time lagged covariance matrices, etc.). However, most of them are highly sensitive to outliers (especially algorithms that employ higher-order statistics). In image processing, outliers may be produced, for example, by faulty pixels in charge-coupled device sensor chips.

Recently, Harmeling et al. (accepted in Neurocomputing) proposed an outlier detection method based on indices that sort data from very typical points (inliers) to very untypical points (outliers). A simple strategy to robustify existing algorithms is to use such indices for outlier rejection. This is indeed possible, but here we show that an appropriately defined outlier index can be used *directly* to solve the ICA problem: since the columns of mixing matrix A are defined only up to sign and scaling, the ICA solution is solely characterized by a

Some of the ideas and experimental results have been presented earlier by Harmeling et al.(submitted) and Harmeling (2004).

Correspondence to: Klaus-Robert Müller; e-mail: klaus@first.fhg.de

Contract grant sponsors: This research has been partly supported by the EU-PASCAL network of excellence (I/st-2002-506778) and DFG (SFB 618-B4).

*The source signals can be recovered only up to scaling and permutation, since a scalar factor can be exchanged between each source and the corresponding column of A without changing $x(t)$. The numbering of the sources (and the columns of A) has no physical interpretation and is nothing but a notational device.

†Also called underdetermined.

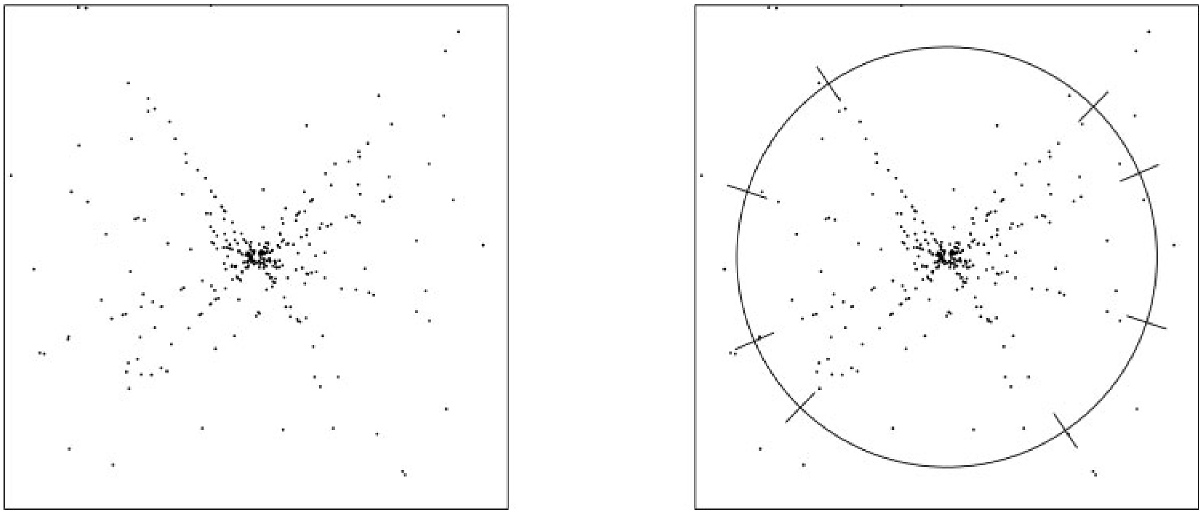

Figure 1. The left panel shows a scatterplot of a 2D mixture of four super-Gaussian source signals. The right panel shows additionally the directions of the points of highest density on the unit circle. Those directions correspond to the columns of *A*.

set of 1D linear subspaces through the origin. For super-Gaussian signals, these subspaces correspond to directions to high point density. Therefore, the idea is to look for 'inliers' in the 'space of directions', and use those data points directly as estimates for the ICA directions (i.e. columns of the mixing matrix A). Figure 1 shows a scatter plot of a 2D mixture of four super-Gaussian source signals (left and right panel). The columns of the mixing matrix are clearly visible as directions in the data space with higher density (right panel). To find these directions, we define an outlier index that sorts the data points from dense ('inlier') to sparse ('outlier') in the 'space of directions'. The inlier points are estimates for the columns of A.

The usual invariances of the ICA problem (scaling and permutation) suggest an outlier index γ, which fulfills the following:

- γ must be invariant under rescaling, i.e. $\gamma(\alpha v) = \gamma(v)$ for $\alpha > 0$,
- γ must be invariant under inversion, i.e. $\gamma(-v) = \gamma(v)$,

with $v \in \mathbb{R}^N$ and $\alpha \in \mathbb{R}_+$. In other words, 'dense' or 'sparse' should be defined with respect to a distance measure between the *directions* of the data points (i.e. angle distance). In the next section, we define such an outlier index and employ it to obtain a robust ICA algorithm.

II. ALGORITHM

This section describes the inlier-based ICA (IBICA) algorithm, which is related to the family of geometric approaches to ICA (see Puntonet et al., 1995; Bofill and Zibulevsky, 2001). The main innovation of IBICA is its usage of an inlier index that makes it particularly robust and allows it to be used in high dimensions.

Step 1: Project the data on the unit sphere. Project all data points $x(1), \ldots, x(T)$ (we assume mean zero) onto the unit sphere by normalizing to length one,

$$z(t) = \frac{x(t)}{\sqrt{x(t)^{\mathrm{T}} x(t)}} = \frac{x(t)}{|x(t)|}.$$

This step ensures the scaling invariance; the distances between the points $z(t)$ on the unit sphere do not depend on the scaling of the original points $x(t)$ but only on the directions. The ICA directions are now given by the dense regions on the sphere. Note that some fraction of the points at the ball around zero is removed, because in noisy settings these points do not contain much information about the correct signal directions (and we avoid division by zero for points exactly from the origin).

Step 2: Calculate γ for an inversion invariant distance. The natural distance measure (angle distance) between two normalized points a and b is the geodesic distance on the unit sphere, but we will use a distance measure based on the Euclidean distance, since it is easier to calculate and yields similar results,[‡]

$$d(a, b) = \min(|a - b|, |a + b|).$$

This distance is invariant under the inversion operation (which maps a vector v onto $-v$). This is the natural distance measure to use for our problem, since we are not interested in the orientation of a vector.

Let now $nn_1(z), \ldots, nn_k(z)$ be the k nearest neighbors of z according to the distance d. We call the average distance of z to its k nearest neighbors the γ index of z, i.e.

$$\gamma(z) = \frac{1}{k} \sum_{j=1}^{k} d(z, nn_j(z)).$$

Intuitively speaking, $\gamma(z)$ is large if z lies in a sparse region (z is probably an outlier), and $\gamma(z)$ is small if z lies in a dense region. The data points with the smallest γ are good candidates for the directions of the signals, i.e. for the columns of A. We call these points inliers.

Step 3: Pick the columns of the mixing matrix. To form the mixing matrix A, we could pick those m data points with the smallest γ values and stack them together. This approach is guaranteed to find only

[‡]For two points a and b on the unit sphere ($|a| = |b| = 1$), the geodesic distance is the angle between those vectors, i.e. $\arccos(a^{\mathrm{T}} b)$. However, for small angles this distance is proportional to the Euclidean distance, $|a - b| = \sqrt{(a-b)^{\mathrm{T}}(a-b)}$, and the relationship is monotonic, i.e. $\arccos(a^{\mathrm{T}} b) < \arccos(a^{\mathrm{T}} c) \iff |a - b| < |a - c|$ for another unit vector c.

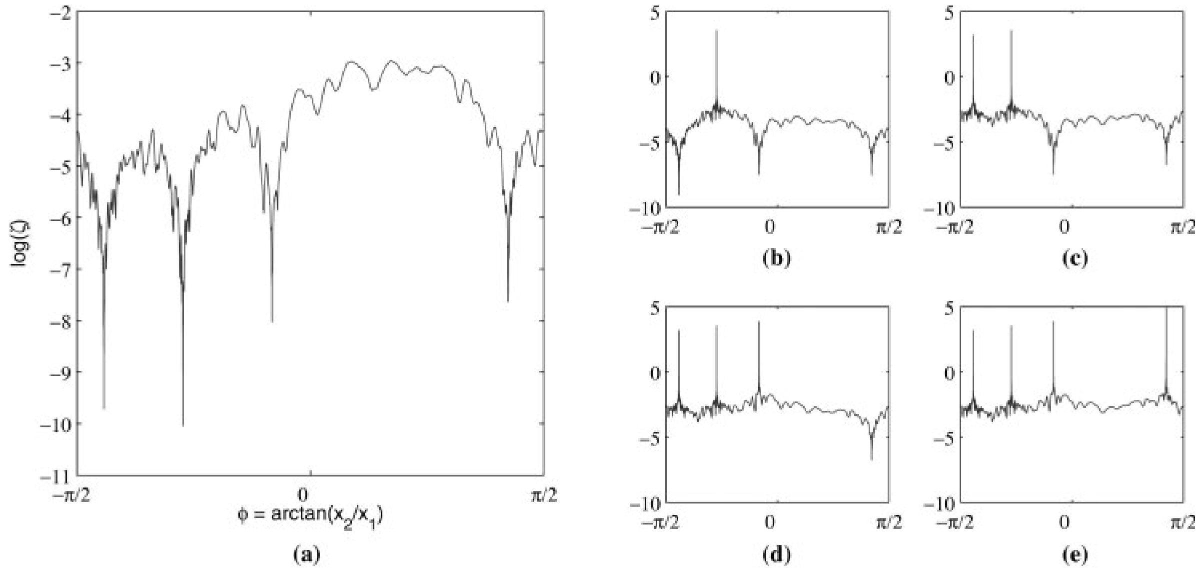

Figure 2. The ζ-landscape of a four-in-two mixture (four sources, two mixtures) plotted as a function of the angle. (a) Shows the initial values, (b) through (e) the values in the next iteration steps. Note that once a direction is chosen as a column of the mixing matrix, the corresponding minimum is removed. This prevents us from taking more than one data point from each direction. After four iterations (e) no expressed minimum is left, which can be used as terminating condition in situations where the true number of sources is unknown.

the first true direction. However, the other $m-1$ estimated directions (i.e. the next lowest ζ values) might belong all to the same column of the true A (i.e. the same peak in Fig. 2a). This problem is solved by iteratively defining several indices, $\zeta^{(1)}$, $\zeta^{(2)}, \ldots$, based on γ that avoid picking directions similar to directions already chosen.

Denote the columns of A by $a_1, \ldots, a_m \in \mathbb{R}^n$. The first column of A is the vector $z \in \mathbb{Z} = \{z(1), \ldots, z(T)\}$ with the smallest γ. Thus, defining for $z \in \mathbb{Z}$

$$\zeta^{(1)}(z) := \gamma(z), \tag{2}$$

the first column vector of A is

$$a_1 := \arg\min_{z \in \mathbb{Z}} \zeta^{(1)}(z). \tag{3}$$

To avoid choosing a direction similar to a_1, we define a new index $\zeta^{(2)}$ that penalizes directions close to a_1,

$$\zeta^{(2)}(z) := \frac{\gamma(z)}{\mathrm{dist}(a_1, z)}. \tag{4}$$

Accordingly, the second column of A is

$$a_2 := \arg\min_{z \in \mathbb{Z}} \zeta^{(2)}(z). \tag{5}$$

Next, we want to avoid the directions of a_1 and a_2 simultaneously, so we define

$$\zeta^{(j)}(z) := \frac{\gamma(z)}{\min_{i<j} \mathrm{dist}(a_i, z)}, \tag{6}$$

which is the general definition of $\zeta^{(j)}$ for $j > 1$. Similarly, we define the jth column of A to be

$$a_j := \arg\min_{z \in \mathbb{Z}} \zeta^{(j)}(z). \tag{7}$$

Following this recipe, we iteratively determine the columns of A. Note that for $j > 1$ we have

$$\zeta^{(j)}(z) \leq \max\left(\zeta^{(j)}(z), \frac{\gamma(z)}{\mathrm{dist}(a_j, z)}\right) = \zeta^{(j+1)}(z) \tag{8}$$

and thus

$$\min_{z \in \mathbb{Z}} \zeta^{(j)}(z) \leq \min_{z \in \mathbb{Z}} \zeta^{(j+1)}(z). \tag{9}$$

Denoting the smallest value of each iteration by

$$\zeta_j := \min_{z \in \mathbb{Z}} \zeta^{(j)}(z) = \zeta^{(j)}(a_j), \tag{10}$$

we see that for $j > 1$ these values increase monotonically,§

$$\zeta_2 \leq \zeta_3 \leq \cdots \tag{11}$$

If all true source directions have been picked—say m directions—we can expect that ζ_{m+1} is much larger than ζ_m, because for all data points with small γ, a direction close to them has been already chosen and there is no expressed minimum in the ζ-landscape left (see Fig. 2). Therefore, a large step in ζ_2, $\zeta_3, \ldots$ determines the number of sources. In practice this works well if there are enough data points.

A. Computational Costs. Step 1 of IBICA—the normalization step—costs $O(nT)$ with n being the dimensionality of the observed data and T the number of data points. For step 2, the whole distance matrix needs to be calculated, costing $O(nT^2)$. Finding the kth nearest neighbor of one point can be done in linear time (selection in expected linear time, see Cormen et al., 1989). γ requires all k nearest neighbors, which can be found (using k times selection in

§Note that ζ_1 is not necessarily smaller than ζ_2: suppose for some $z \in \mathbb{Z}$ we have $\zeta_1 = \gamma(a_1) < \rho\gamma(a_1) = \gamma(z)$ with $1 < \rho < \sqrt{2}$. Assume further that z has maximal distance to a_1 on the unit sphere with respect to dist, i.e. dist $(a_1, z) = \sqrt{2}$. Then we have $\zeta^{(2)}(z) = \rho\gamma(a_1)/\sqrt{2} < \gamma(a_1) = \zeta_1$ and thus $\zeta_2 < \zeta_1$.

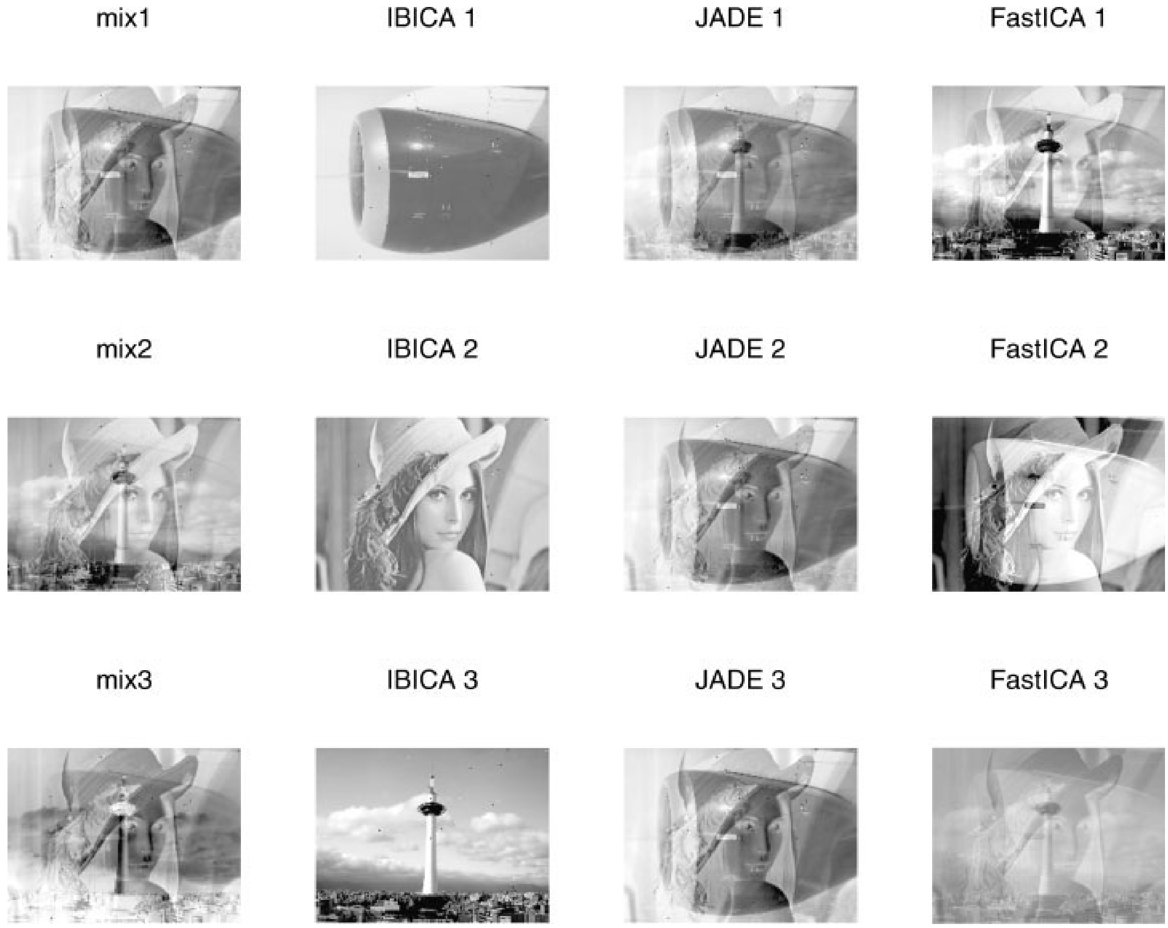

Figure 3. Separation of mixtures with pixel defects (outliers). The first column shows the mixtures, the second, third and fourth column the reconstructed source images as estimated by IBICA, JADE and FastICA, respectively.

expected linear time) in $O(kT)$ for each point, i.e. in total $O(T^2n + T^2k)$ for all points. For large k, we can also sort all neighbors of a point (in $O(T \log T)$), i.e. in total $O(T^2n + T^2 \log T)$ for all points. Summarizing, step 2 requires $O(T^2n + T^2 \max(k, \log T))$. To simplify, we can say that the time complexity of step 2 is $O(T^2 \log T)$. Picking one direction in step 3 requires to sort the current index (in $O(T \log T)$), to calculate the distances to the last chosen direction (in $O(nT)$) and to update the indices (in $O(T)$). So, in total, IBICA requires $O(T^2 \log T)$.

The computational costs of step 2 can be dramatically reduced by randomly partitioning the data into smaller disjoint subsets of approximately equal size and calculating for each subset the requested index. The scale of the indices changes with the number of data points, i.e. limiting the calculation to a subset, increases the scale compared to considering all data points. However, since all subsets have equal size, the resulting indices are comparable. Note that the smaller the subsets are chosen, the coarser the indices will be.

III. EXPERIMENTS

In this section, we compare our algorithm (IBICA) to JADE (Cardoso and Souloumiac, 1993) and FastICA (Hyvarinen and Oja, 1997), which are well-established algorithms for standard ICA, and to sparsenet (Olshausen and Field, 1996, 1997), hardLOST (O'Grady and Pearlmutter, 2004a) and softLOST (O'Grady and Pearlmutter, 2004b), which are algorithms for overcomplete ICA. First, we show qualitatively the ability of IBICA to cope with outliers in an image separation task and then we investigate the performance of our algorithm more comprehensively in controlled experiments. Finally, we demonstrate that IBICA can solve the overcomplete image separation problem.

Note that all runs of IBICA used $k = 20$. Our experience has shown that the choice of k is uncritical and the results for different k are similar to the ones presented.

A. Image Separation in the Presence of Outliers. In our first experiment, we linearly mixed three different images and added some outliers to these mixtures. The pictures each consist of 307,200 pixels (640×480); 200 pixels (0.07%) have been replaced by outliers, i.e. points with (Gaussian) random values of a 10 times greater standard deviation than the original points. These mixtures with pixel defects can be seen in the first column of Figure 3. However, they are neither sparse nor very super-Gaussian. So, to separate the source images with IBICA, it is necessary to *sparsify* them by a suitable transformation. This is achieved by a wavelet transformation using a Haar wavelet from the MATLAB Wavelet Toolbox. The ICA problem is then solved for the wavelet coefficients, and after that the inverse wavelet transformation yields the source estimates (in the standard ICA setting ($M = N$), we might as well just estimate the unmixing matrix on the wavelet coefficients and apply it directly to the mixtures). Column 2 of Figure 3 shows that IBICA is able to find the correct source images. However, if the mixtures are presented to JADE or FastICA, the outliers in the data set will make them fail, regardless of whether the mixtures have been sparsified or not.

B. Performance Measures. To investigate the performance of IBICA more systematically, we use a performance measure that applies to the standard ICA and to overcomplete ICA. Assume that the mixing matrix A and its estimator $\hat{A}$ are column normalized (i.e. the norm of the columns of these matrices is one). We then define:

$$\mathrm{pm}(A,\hat{A}) = 1 - \left(\frac{1}{2M}\sum_{i=1}^{M}\max_j |A^{\mathrm{T}}\hat{A}|_{ij} + \frac{1}{2M}\sum_{j=1}^{M}\max_i |A^{\mathrm{T}}\hat{A}|_{ij}\right).$$

The idea is to match each column vectors of A with one of $\hat{A}$ and vice versa. The performance measure is symmetrical (pm(A, $\hat{A}$) = pm($\hat{A}$, A)), smaller or equal to one and zero only if $\hat{A} = AP$ with P being a permutation matrix (i.e. perfect solution). The main advantage of this performance measure is that it not only applies to standard ICA but also to the overcomplete case, because it is not restricted to square mixing matrices.

C. Robustness Against Outliers. In the experiments of this and the following sections, we produce super-Gaussian source signals by taking Gaussian noise to the power of five. We mix those sources by randomly chosen squared mixing matrices. FastICA is used in two variants: first with ‘pow3’ non-linearity and stabilization turned off, and second with the more robust ‘tanh’ non-linearity and stabilization turned on. To help FastICA further against IBICA, which is by definition a robust method, we provide for both versions of FastICA the true whitening matrix. The performance of these versions of FastICA is guaranteed to be better than FastICA methods that use a robust estimation of the whitening matrix. The results show that, without outliers, the performances of IBICA and FastICA are all excellent. To test for outlier-robustness, we replace varying proportions of the data with outliers, i.e. with Gaussian distributed data points with very large variance $\sigma^2 = 10{,}000$.

Figure 4 reveals that the performance of the robustified versions of FastICA decreases rapidly, probably because outliers create directions of high kurtosis, which are attractive for algorithms that use higher-order statistics. This is in contrast to IBICA, which shows perfect performance up to 20% outliers for the 2D data (upper panel of Fig. 4) and up to 10% outliers for the 10D data (lower panel of Fig. 4), thus providing to be extremely robust against outliers.

Note that in this and the following experiments, the curves show the median performances over 100 runs (if not stated differently in the caption) because the separation performance of the algorithms depend strongly on the actual realization of the noise. The signals presented to the different algorithms are always the same.

D. Robustness Against Super-Gaussian Noise. In the next experiment, we spoil the data by adding kurtotic noise with varying standard deviations to the mixtures. Figure 5 shows the evolution of the performance index as a function of the standard deviation of the kurtotic noise (we used multi-dimensional Gaussian noise, where we change the absolute value to the power of five) in two dimensions and in 10 dimensions. Again the robustified versions of Fast ICA are inferior to IBICA. In the 2D case (upper panel of Fig. 5), this difference is highlighted by the fact that up to a standard deviation of 0.4 IBICA performs almost perfectly. Also in 10D example (lower panel of Fig. 5), IBICA outperforms the robustified versions of Fast ICA. Again, we show the median over 100 runs.

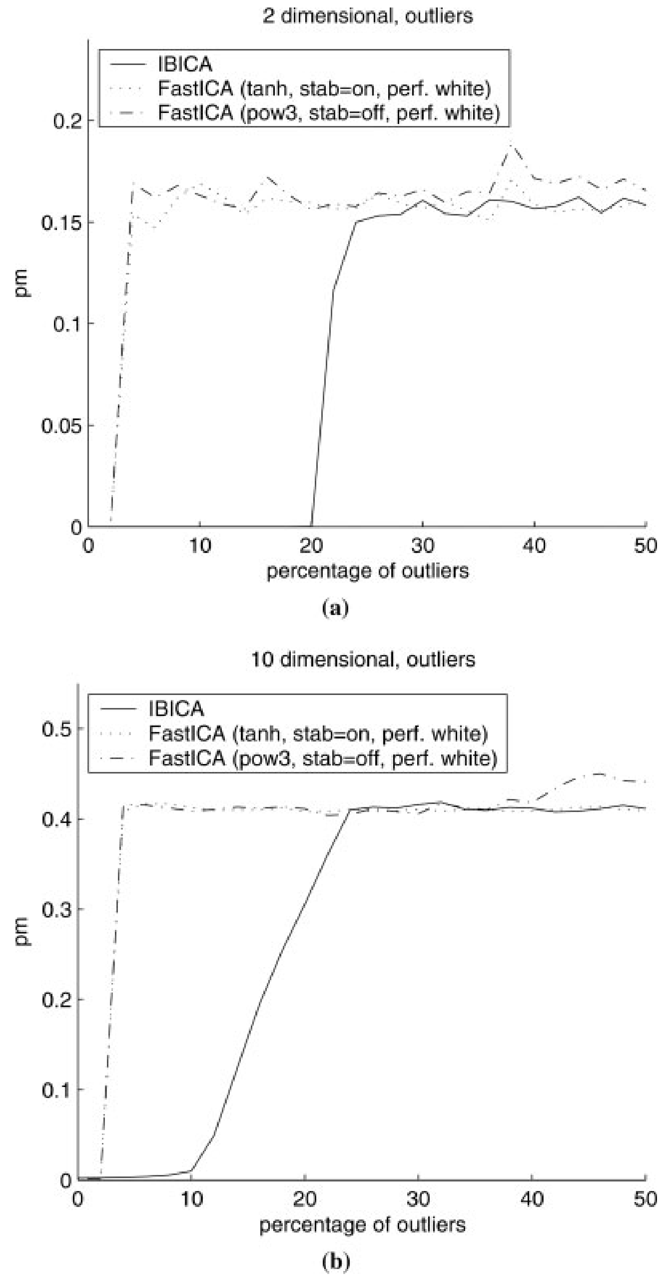

Figure 4. Performance-index vs. outlier level in 2D (a) and 10D mixtures (shown is the median of 100 runs).

E. Robustness Against Outliers for Overcomplete ICA. IBICA is also applicable to the overcomplete setting. Thus we can compare its performance with respect to robustness against other algorithms for overcomplete ICA. In particular, we test against sparsenet (Olshausen and Field, 1996, 1997), hardLOST (O’Grady and Pearlmutter, 2004a) and softLOST (O’Grady and Pearlmutter, 2004b). The sources are generated as in the previous sections.

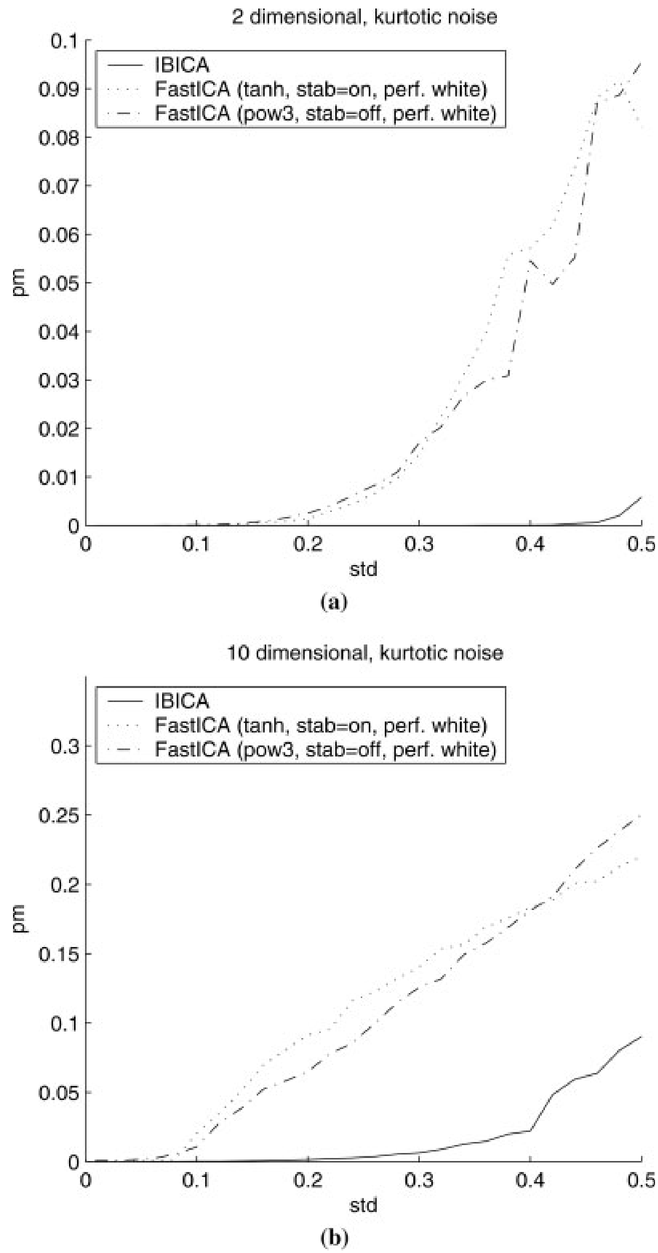

Figure 5. Performance-index vs. standard deviation of the kurtotic noise in 2D (a) and 10D mixtures (shown is the median of 100 runs).

The mixing matrices are no longer squared but have sizes 2×3 and 10×16, respectively.

Figure 6 shows the results: without outliers all methods show comparable performance. However, the upper panel demonstrates that IBICA is also in the overcomplete setting able to find the true solutions for 2D even if up to 17% of the data have been replaced by outliers. The other method fails as soon as there are outliers. The lower panel confirms this finding also for higher dimensional data (10D). Again, IBICA is superior to the other methods and is able to find the true directions for up to 5% outliers.

F. Overcomplete Mixture of Images. Finally, we apply IBICA to a 2D overcomplete mixture of three source images (Fig. 7, left column). IBICA has to estimate a 2×3 mixing matrix. Applying IBICA to the sparsified data (again using a Haar wavelet), it is able to estimate the correct mixing matrix (pm 10^{-4}).

Of course, to reconstruct the source signals in an overcomplete setting, it is not enough to estimate the mixing matrix. The wavelet coefficients (Fig. 7, lower left corner) have to be assigned to the different ICA directions. However, as long as the source signals are not perfectly sparse (or, to be more precise: disjoint), this assignment is not unique, so the source reconstruction will never be lossfree. There are different heuristics how the wavelet coefficients can be assigned to the different sources. The simplest would be a winner-take-all strategy (i.e. each coefficient is uniquely assigned to one source by just projecting it to the closest ICA direction). In our

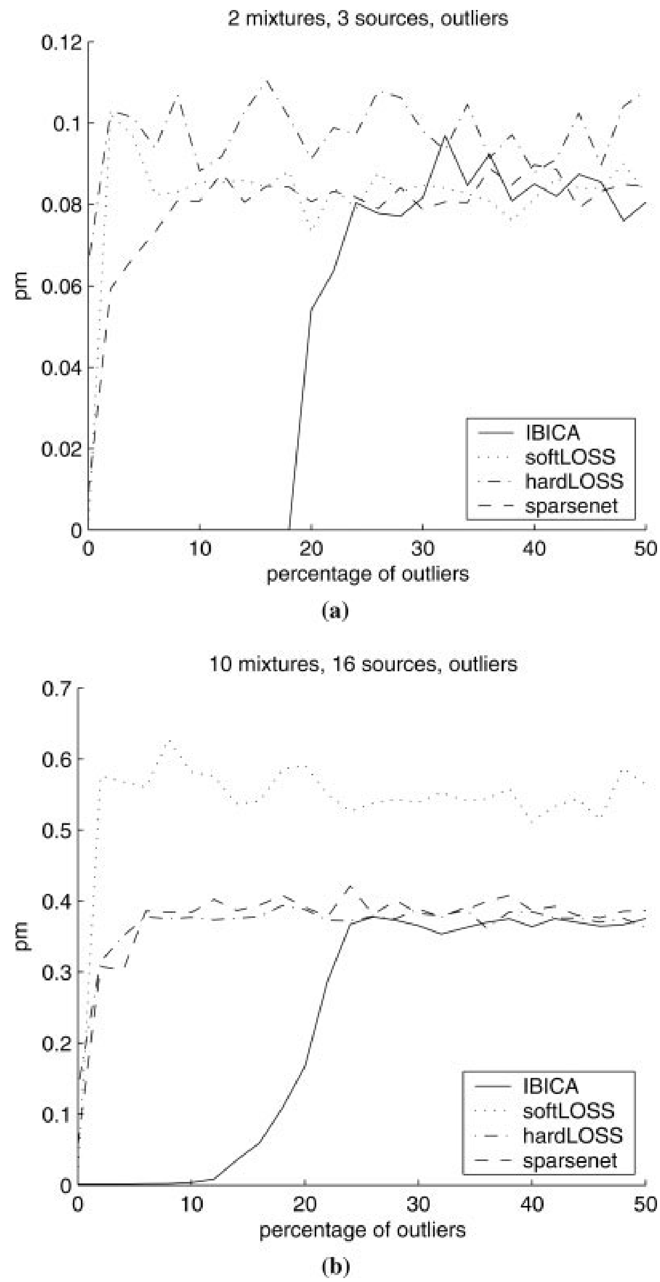

Figure 6. Performance-index vs. outlier level in 2D (a) and 10D mixtures (shown is the median of 100 runs (upper panel) and 8 runs (lower panel)).

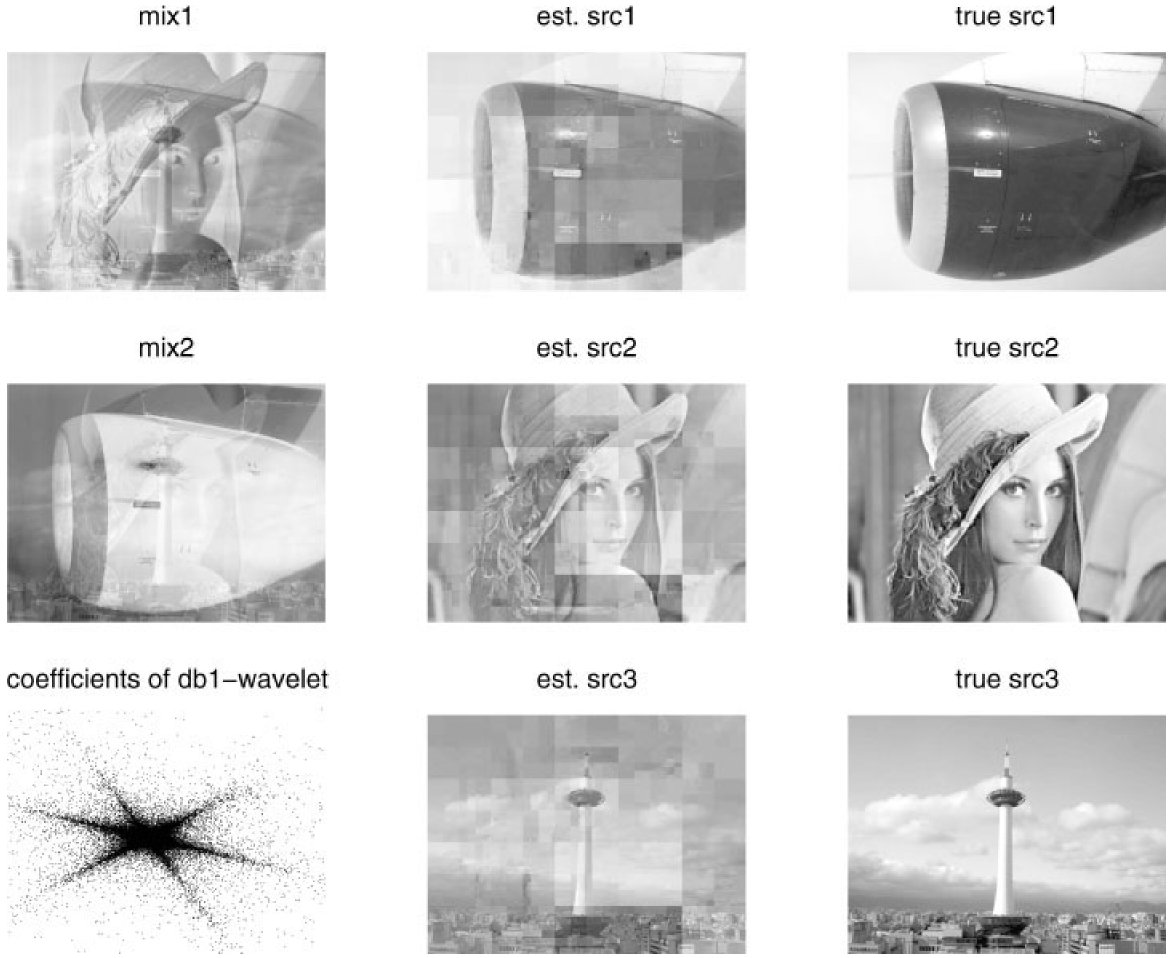

Figure 7. Separation of an overcomplete mixture by IBICA. The left column shows the two observed mixtures and a scatter plot of the wavelet coefficients, the middle column the estimated sources and the right column the true sources for comparison. Note that although the sources can be separated well, the picture reconstruction is not loss-free and shows typical artifacts of the used wavelet.

experiment, we used a more sophisticated method proposed by Bofill and Zibulevsky (2001). Here, each coefficient is decomposed as linear combination of the two closest source signals. The reconstructed sources are shown in the middle column of Figure 7. Although the images are well separated, the reconstruction loss is clearly visible as typical artifacts produced by the Haar wavelet. Note that there might be better choices for the sparsifying transformation (i.e. that sparsify better or whose artifacts are less disturbing), but in this paper we mainly focus on the principle of estimating the ICA direction.

IV. CONCLUSION

Obtaining robust meaningful decompositions is essential when applying blind source separation techniques to data from the real world (see e.g. Meinecke et al., 2002). In many applications, the data is strongly contaminated with measurement noise and outliers where unusual events not belonging to the probability distribution of interest or non-standard noise are measured. Such outlier events pose a severe problem to most existing ICA algorithms, especially the ones that optimize kurtosis-based indices. Our contribution—besides pointing out this fundamental issue—is to use ‘inlier’ data points directly as estimates for the columns of the mixing matrix. Since IBICA is indifferent to outliers, it is an extremely outlier-robust method for standard and overcomplete ICA as we have shown in the experiments sections, that obviously can also be used if there are no outliers. Furthermore, because this novel framework for ICA does not depend on the dimensionality of the problem, it can be readily used also in overcomplete/underdetermined scenarios. Simulations and applications to image separation problems underline these insights. Future research will continue the quest for more robust blind source separation algorithms that can have a wider practical applicability.

ACKNOWLEDGMENTS

The authors thank Andreas Ziehe, Motoaki Kawanabe and Christin Schäfer for valuable discussions.

REFERENCES

P. Bofill and M. Zibulevsky, Underdetermined blind source separation using sparse representations, Signal Process 81 (2001), 2353–2362.

J.-F. Cardoso and A. Souloumiac, Blind beamforming for non Gaussian signals, IEE Proc-F 140(6) (1993), 362–370.

S.S. Chen, D.L. Donoho, and M.A. Saunders, Atomic decomposition by basis pursuit, SIAM J Sci Comput 20(1) (1998), 33–61.

T.H. Cormen, C.E. Leiserson, and R.L. Rivest, Introduction to algorithms, MIT Press, Cambridge, MA, 1989.

S. Harmeling, Independent component analysis and beyond, Ph.D. Thesis, Universität Potsdam, Oct 2004.

S. Harmeling, G. Dornhege, D. Tax, F. Meinecke, and K.-R. Müller, From Outliers to Prototypes: Ordering Data, available at http://ida.first.fhg.de/∼harmeli/ordering.pdf, accepted for publication in Neurocomputing.

A. Hyvarinen, J. Karhunen, and E. Oja, Independent component analysis, Wiley, New York, 2001.

A. Hyvärinen and E. Oja, A fast fixed-point algorithm for independent component analysis, Neural Comput 9(7) (1997), 1483–1492.

T-W. Lee, M.S. Lewicki, M. Girolami, and T.J. Sejnowski, Blind source separation of more sources than mixtures using overcomplete representations, IEEE Signal Process Lett 6(4) (1999), 78–90.

F. Meinecke, S. Harmeling, and K.-R. Müller, Robust ICA for super-Gaussian sources, In C.G. Puntonet and A. Prieto (Editors), Lecture Notes Computer Science, Vol. 3195, Springer, Berlin, 2004, pp. 217–224, Proc Int Workshop on Independent Component Analysis and Blind Signal Separation (ICA2004).

F. Meinecke, A. Ziehe, M. Kawanabe, and K.-R. Müller, A resampling approach to estimate the stability of one-dimensional or multidimensional independent components. IEEE Trans Biomed Eng, 49 (2002), 1514–1525.

P.D. O'Grady and B.A. Pearlmutter, Hard-lost: Modified *k*-means for oriented lines, Proc Irish Signals Systems Conf, 2004a, pp. 247–252.

P.D. O'Grady and B.A. Pearlmutter, Soft-lost: Em on a mixture of oriented lines, In C.G. Puntonet and A. Prieto (Editors), Lecture Notes in Computer Science, Vol. 3195, Springer, Berlin, 2004b, pp. 428–435, Proc Int Workshop on Independent Component Analysis and Blind Signal Separation (ICA2004).

B.A. Olshausen and D.J. Field, Emergence of simple-cell receptive field properties by learning a sparse code for natural images, Nature 381 (1996), 607–609.

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

C.G. Puntonet, A. Prieto, C. Jutten, M. Rodriguez-Alvarez, and J. Ortega, Separation of sources: A geometry-based procedure for reconstruction of *n*-valued signals, Signal Process 46 (1995), 267–284.

M. Zibulevsky and B.A. Pearlmutter, Blind source separation by sparse decomposition in a signal dictionary, Neural Comput 13 (2001), 863–882.