IEEE Transactions on Signal Processing, 52(5):1250-1256 (2004).

CuBICA: Independent Component Analysis by Simultaneous Third- and Fourth-Order Cumulant Diagonalization

Tobias Blaschke and Laurenz Wiskott

***Abstract*— CuBICA, an improved method for independent component analysis (ICA) based on the diagonalization of cumulant tensors is proposed. It is based on Comon's algorithm [1] but it takes third- and fourth-order cumulant tensors into account simultaneously. The underlying contrast function is also mathematically much simpler and has a more intuitive interpretation. It is therefore easier to optimize and approximate. A comparison with Comon's and three other ICA-algorithms on different data sets demonstrates its performance.**

***Index Terms*— Independent Component Analysis, Cumulant, Contrast**

I. Introduction

A vectorial signal $\mathbf{x}(t) = [x_1(t), \ldots, x_N(t)]^T$ to be analyzed, which we will refer to as input signal, is often a mixture of some underlying signal components $s_i(t)$ coming from different sources. For instance, the sound we hear is usually a superposition of several sound sources, such as a person speaking and a phone ringing. In a simple model of this data generation process, it is assumed that there are as many sources as input signal components, that the source signal components s_i (person and phone) are statistically independent, that at most one s_i is normally distributed, and that the mixing is linear and noise free, yielding the relation

$$\mathbf{x} = \mathbf{A}\mathbf{s}\,, \tag{1}$$

with fixed mixing matrix $\mathbf{A}$ and source signal $\mathbf{s}(t) = [s_1(t), \ldots, s_N(t)]^T$. In the following we will drop the reference to time, and simply assume some sets of source and input data related by (1). We also assume for simplicity that the source signal and input signal have zero mean.

The input components are usually statistically dependent, due to the mixing process, while the sources are not. If one succeeds in finding a matrix $\mathbf{R}$ that yields statistically independent output components u_k, given by $\mathbf{u} = \mathbf{R}\mathbf{x} = \mathbf{R}\mathbf{A}\mathbf{s}$, one can recover the original sources s_i up to a permutation and constant scaling of the sources. $\mathbf{R}$ (or sometimes $\mathbf{Q}$, see below) is called the unmixing matrix and finding the matrix is referred to as independent component analysis (ICA).

To fix the arbitrary scaling factors, we require the output components to have unit variance. For technical convenience, the unmixing is performed in two steps. First a whitening matrix $\mathbf{W}$ is applied, yielding the whitened signal $\mathbf{y} = \mathbf{W}\mathbf{x}$, then an orthogonal rotation matrix $\mathbf{Q}$ is applied yielding the estimated source signal

$$\mathbf{u} = \mathbf{Q}\mathbf{y} = \underbrace{\mathbf{Q}\mathbf{W}}_{=\mathbf{R}}\mathbf{x} = \underbrace{\mathbf{Q}\mathbf{W}}_{=\mathbf{R}}\mathbf{A}\mathbf{s} \tag{2}$$

as an output signal.

There is a variety of methods for performing ICA and a large body of literature (see [2]–[4] for an overview). Well known examples are the Infomax approach by [5], FastICA, a fixed-point algorithm by [6], and the cumulant based methods [1], [7], [8]. We extend here on the latter and present an improved algorithm that takes third- and fourth-order cumulants into account simultaneously and, at the same time, is simpler and faster than Comon's algorithm [1], which our algorithm is based upon. The algorithm is described in Section II and a performance comparison is given in Section III. We conclude with a brief discussion in Section IV. All simulations were done with Matlab; analytical calculations were supported by Mathematica.

II. Improved ICA Algorithm

A. Cumulants and Independence

Statistical properties of the output data set $\mathbf{u}$ can be described by its moments or, more conveniently, by its cumulants $C^{(\mathbf{u})}_{\cdots}$. Since the data have zero mean, the sample cumulants up to order four read

$$\begin{aligned}
C^{(\mathbf{u})}_{i} &= 0\,, \\
C^{(\mathbf{u})}_{ij} &= \langle u_i u_j \rangle\,, \\
C^{(\mathbf{u})}_{ijk} &= \langle u_i u_j u_k \rangle\,, \\
C^{(\mathbf{u})}_{ijkl} &= \langle u_i u_j u_k u_l \rangle - \langle u_i u_j \rangle \langle u_k u_l \rangle \\
&\quad - \langle u_i u_k \rangle \langle u_j u_l \rangle - \langle u_i u_l \rangle \langle u_j u_k \rangle\,,
\end{aligned} \tag{3}$$

with $\langle\cdot\rangle$ indicating the mean over all data points. Higher order cumulants are not considered here. Cumulants of a given order form a tensor. The diagonal elements characterize the distribution of single components. For example, $C^{(\mathbf{u})}_{i}$, $C^{(\mathbf{u})}_{ii}$, $C^{(\mathbf{u})}_{iii}$, and $C^{(\mathbf{u})}_{iiii}$, the autocumulants of first to fourth order in $\mathbf{u}$, are the mean, variance, skewness, and kurtosis of u_i, respectively. The off-diagonal elements or cross-cumulants (all cumulants with $ijkl \neq iiii$) characterize the statistical dependencies between components. If and only if all components u_i

This work has been supported by the Volkswagen Foundation through a grant to LW for a junior research group.

T. Blaschke and L. Wiskott are with the Institute for Theoretical Biology, Humboldt University, Berlin (e-mail:{t.blaschke,l.wiskott}@biologie.hu-berlin.de).

are statistically independent, the off-diagonal elements vanish and the cumulant tensors (of all orders) are diagonal (assuming infinite amount of data).

Thus, ICA is equivalent to finding an unmixing matrix that diagonalizes the cumulant tensors $C^{(\mathbf{u})}_{\dots}$ of the output data u_i, at least approximately. The first order cumulant tensor is a vector and does not have off-diagonal elements. The second order cumulant tensor can be diagonalized easily by whitening the input data $\mathbf{x}$ with an appropriate matrix $\mathbf{W}$, yielding $\mathbf{y} = \mathbf{W}\mathbf{x}$ with $\langle \mathbf{y}\mathbf{y}^T \rangle = \mathbf{I}$ ($\mathbf{I}$ denotes the N-dimensional identity matrix). In general $\mathbf{W}$ also fulfills the condition

$$\mathbf{W}\mathbf{A}\mathbf{A}^T\mathbf{W}^T = \mathbf{I}\,, \tag{4}$$

assuming the source signal components have unit variance. It can be shown that this exact diagonalization of the second order cumulant tensor of the whitened data $\mathbf{y}$ is preserved if and only if the final matrix $\mathbf{Q}$ generating the output data $\mathbf{u} = \mathbf{Q}\mathbf{y}$ is orthogonal, i.e. a pure rotation possibly plus reflections [9]. In general there is no orthogonal matrix that would diagonalize also the third- or fourth-order cumulant tensor, thus the diagonalization of these tensors can only be done approximately and we need to define an optimization criterion for this approximate diagonalization, which is done in the next section.

B. Contrast Function

In order to formalize the approximate diagonalization of the cumulant tensors of order three and four we define the following criterion

$$\begin{aligned}\bar{\Psi}_{34}(\mathbf{u}) := & \frac{1}{3!}\sum_{\alpha\beta\gamma\neq\alpha\alpha\alpha}\left(C^{(\mathbf{u})}_{\alpha\beta\gamma}\right)^2 + \\ & \frac{1}{4!}\sum_{\alpha\beta\gamma\delta\neq\alpha\alpha\alpha\alpha}\left(C^{(\mathbf{u})}_{\alpha\beta\gamma\delta}\right)^2\,,\end{aligned} \tag{5}$$

which is simply the sum over the squared third- and fourth-order off-diagonal elements and needs to be minimized. The factors $\frac{1}{3!}$ and $\frac{1}{4!}$ arise from an expansion of the Kullback-Leibler divergence in $\mathbf{u}$, which provides a rigorous derivation of this criterion [7], [10].

Since the square sum over all elements of a cumulant tensor is preserved under any orthogonal transformation $\mathbf{Q}$ of the underlying data $\mathbf{y}$ [11], one can equally well maximize the sum over the diagonal elements,

$$\Psi_{34}(\mathbf{u}) := \frac{1}{3!}\sum_{\alpha}\left(C^{(\mathbf{u})}_{\alpha\alpha\alpha}\right)^2 + \frac{1}{4!}\sum_{\alpha}\left(C^{(\mathbf{u})}_{\alpha\alpha\alpha\alpha}\right)^2\,, \tag{6}$$

instead of minimizing the sum over the off-diagonal elements (5). $\Psi_{34}(\mathbf{u})$ is obviously much simpler than $\bar{\Psi}_{34}(\mathbf{u})$. Notice that this is a contrast as defined by [7] because all functionals $\sum_\alpha \left(C^{(\mathbf{u})}_{\alpha\alpha\dots\alpha}\right)^2$ of cumulants of order ≥ 2 are contrasts and their sum $\Psi_{34}(\mathbf{u})$ is a contrast, too [12]. For a more general approach to contrast functions see [13].

Due to the multilinearity of the cumulants $C^{(\mathbf{u})}_{\dots}$ in $C^{(\mathbf{y})}_{\dots}$, (6) can be rewritten as

$$\begin{aligned}\Psi_{34}(\mathbf{Q},\mathbf{y}) = & \frac{1}{3!}\sum_{\alpha}\Bigg(\underbrace{\sum_{\beta\gamma\delta} Q_{\alpha\beta}Q_{\alpha\gamma}Q_{\alpha\delta}C^{(\mathbf{y})}_{\beta\gamma\delta}}_{C^{(\mathbf{u})}_{\alpha\alpha\alpha}}\Bigg)^2 \\ & + \frac{1}{4!}\sum_{\alpha}\Bigg(\underbrace{\sum_{\beta\gamma\delta\epsilon} Q_{\alpha\beta}Q_{\alpha\gamma}Q_{\alpha\delta}Q_{\alpha\epsilon}C^{(\mathbf{y})}_{\beta\gamma\delta\epsilon}}_{C^{(\mathbf{u})}_{\alpha\alpha\alpha\alpha}}\Bigg)^2 .\end{aligned} \tag{7}$$

$C^{(\mathbf{y})}_{\dots}$ are the cumulants of the whitened data set $\mathbf{y}$ and $Q_{..}$ are the elements of the rotation matrix $\mathbf{Q}$. With $\mathbf{u} = \mathbf{Q}\mathbf{y}$ Equations (6) and (7) are formally related by $\Psi_{34}(\mathbf{u}) = \Psi_{34}(\mathbf{I},\mathbf{u}) = \Psi_{34}(\mathbf{Q},\mathbf{y})$. $\Psi_{34}(\mathbf{Q},\mathbf{y})$ is now subject to an optimization procedure to find the orthogonal matrix $\mathbf{Q}$ that maximizes it.

C. Givens Rotations

A Givens rotation is a rotation around the origin within the plane of two selected components μ and ν and has the matrix form

$$Q^{\mu\nu}_{\alpha\beta} := \begin{cases} \cos(\phi) & \text{for } (\alpha,\beta)\in\{(\mu,\mu),(\nu,\nu)\} \\ -\sin(\phi) & \text{for } (\alpha,\beta)\in\{(\mu,\nu)\} \\ \sin(\phi) & \text{for } (\alpha,\beta)\in\{(\nu,\mu)\} \\ \delta_{\alpha\beta} & \text{otherwise} \end{cases} \tag{8}$$

with Kronecker symbol $\delta_{\alpha\beta}$ and rotation angle ϕ. In contrast function (6) only the cumulants with $\alpha \in \{\mu,\nu\}$ are effected by such a rotation. Any orthogonal $N \times N$ matrix such as $\mathbf{Q}$ can be written as a product of $\frac{N(N-1)}{2}$ (or more) Givens rotation matrices $\mathbf{Q}^{\mu\nu}$ (for the rotation part) and a diagonal matrix with diagonal elements ± 1 (for the reflection part). Since reflections do not matter in our case we only consider the Givens rotations. For simplicity and without loss of generality we now consider only the subspace of two selected components, so that the Givens rotation matrix becomes

$$\mathbf{Q}^{\mu\nu} = \begin{pmatrix} \cos(\phi) & \sin(\phi) \\ -\sin(\phi) & \cos(\phi) \end{pmatrix}\,. \tag{9}$$

Contrast function (7) can then be rewritten as $\Psi_{34}(\phi,\mathbf{y}) = \Psi_3(\phi,\mathbf{y}) + \Psi_4(\phi,\mathbf{y})$ with

$$\begin{aligned}\Psi_n(\phi,\mathbf{y}) := & \frac{1}{n!}\sum_{i=0}^{n} d_{ni}\left(\cos(\phi)^{(2n-i)}\sin(\phi)^{i}\right) \\ & + \frac{1}{n!}\sum_{i=0}^{n} d_{ni}\left(\cos(\phi)^{i}(-\sin(\phi))^{(2n-i)}\right)\end{aligned} \tag{10}$$

with some constants d_{ni} that depend only on the cumulants $C^{(\mathbf{y})}_{\dots}$ before rotation (see Appendix A). To simplify this equation [1] defined some auxiliary variables $\theta := \tan(\phi)$ and $\xi := \theta - \frac{1}{\theta}$ and derived

$$\Psi_3(\theta,\mathbf{y}) = \frac{1}{3!}\left(\theta + \frac{1}{\theta}\right)^{-3}\sum_{i=1}^{3} a_i\left(\theta^i - (-\theta)^{-i}\right)\,, \tag{11}$$

$$\Psi_4(\xi,\mathbf{y}) = \frac{1}{4!}\left(\xi^2 + 4\right)^{-2}\sum_{i=0}^{4} b_i\xi^i \tag{12}$$

for (10), with some constants a_i and b_i depending on the cumulants before rotation. To maximize (11) or (12) one has to take their derivative and find the root giving the largest value for Ψ_3 or Ψ_4, respectively. With this formulation only either the third-order or the fourth-order diagonal cumulants can be maximized but not both simultaneously.

In a more direct approach and after some quite involved calculations using various trigonometric theorems, we were able to derive a contrast function that (i) combines third- and fourth-order cumulants, (ii) is mathematically much simpler, (iii) has a more intuitive interpretation, and (iv) is therefore easier to optimize and approximate. We found

$$\Psi_{34}(\phi, \mathbf{y}) = A_0 + A_4 \cos(4\phi + \phi_4) + A_8 \cos(8\phi + \phi_8) \quad (13)$$

with some constants A_0, A_4, A_8 and ϕ_4, ϕ_8 that depend only on the cumulants $C^{(\mathbf{y})}_{\cdots}$ before rotation (see Appendix B). The third term comes from the fourth order cumulants only while the first two terms incorporate information from the third- and the fourth-order cumulants. Contrast functions for third- or fourth-order cumulants only, i.e. Ψ_3 or Ψ_4, can be easily obtained by setting all fourth- or third-order cumulants to zero, respectively.

It is actually relatively easy to see that it is possible to write the contrast in such a simple form. Firstly, rotation by multiples of $\frac{\pi}{2}$ corresponds to a permutation of the two components possibly plus sign changes, which does not affect the value of the contrast. Therefore, Ψ_{34} has a periodicity of $\frac{\pi}{2}$ and can be written as a sum of cos-functions with frequencies 0, 4, 8, 12, 16, etc. Secondly, the terms in (10) are products of at most eight $\sin(\phi)$ and $\cos(\phi)$ functions, which can lead at most to a frequency of 8. Taking together these two arguments it is clear that only the frequencies 0, 4, and 8 are present and the contrast can be written in the form of (13). Because of the $\frac{\pi}{2}$ periodicity it suffices to evaluate the contrast in the interval $[\phi_4 - \frac{\pi}{4}, \phi_4 + \frac{\pi}{4}]$.

[14] derived a related formula for third-order cumulants only that is quadratic in $\sin(2\phi)$ and $\cos(2\phi)$ and can be transformed to an expression similar to (13).

D. Unmixing Schedule

Unmixing for $N = 2$ can now be achieved in four steps: (i) compute the constants in (13), (ii) find the angle ϕ_{max} that maximizes $\Psi_{34}(\phi, \mathbf{y})$ in (13), (iii) calculate the Givens rotation-matrix $\mathbf{Q}^{\mu\nu}$ according to (9), and (iv) apply it to the whitened signal $\mathbf{y}$ to obtain estimated source signal $\mathbf{u} = \mathbf{Q}^{\mu\nu}\mathbf{y}$. Since Ψ_{34} is maximal, the cumulant tensors $C^{(\mathbf{u})}_{ijkl}$ and $C^{(\mathbf{u})}_{ijk}$ are as diagonal as possible according to contrast (7) and the estimated signal components u_i are maximally statistically independent.

There are different ways to find the angle ϕ_{max} that maximizes $\Psi_{34}(\phi, \mathbf{y})$. Since all constants in Eq. (13) are known and ϕ_{max} has to lie in the interval $\left[\phi_4 - \frac{\pi}{4},\ \phi_4 + \frac{\pi}{4}\right]$ we simply calculated $\Psi_{34}(\phi, \mathbf{y})$ for 1000 equidistant values of ϕ covering this interval and took the angle with largest value. We also tested a Matlab built-in function based on Golden Section search and parabolic interpolation, which was significantly slower, but found no difference in the unmixing performance.

For $N > 2$ the contrast maximization follows directly from the $N = 2$ case. We denote the contrast function for a selected pair μ, ν of components by $\Psi^{\mu\nu}_{34}(\phi^{\mu\nu}, \mathbf{y})$. Note that pairwise statistical independence of the signal components implies mutual independence of all signal components [7]. Therefore it is sufficient to iteratively maximize all $\Psi^{\mu\nu}_{34}$ like in the case of $N = 2$ until $\phi^{\mu\nu}_{max}$ is smaller then a given threshold ϵ for every pair μ, ν. In practice this can take several sweeps through all pairs. Every sweep consists of $N(N-1)/2$ rotations.

After centering and whitening, a maximization schedule for $N > 2$ could be as follows:

(1) Initialize auxiliary variables $\mathbf{Q}' = \mathbf{I_n}$ and $\mathbf{y}' = \mathbf{y}$

(2) Choose a pair of components μ and ν (randomly or in any given order)

(3) Calculate the Cumulants that are needed for $\Psi^{\mu\nu}_{34}(\phi^{\mu\nu}, \mathbf{y}')$

(4) Find the angle $\phi^{\mu\nu}_{max}$ such that $\Psi^{\mu\nu}_{34}(\phi^{\mu\nu}_{max}, \mathbf{y}')$ is maximal

(5) If $\phi^{\mu\nu}_{max} > \epsilon$ update $\mathbf{Q}'$ according to $\mathbf{Q}' \to \mathbf{Q}^{\mu\nu}\mathbf{Q}'$

(6) Rotate the signal components: $\mathbf{y}' \to \mathbf{Q}^{\mu\nu}\mathbf{y}'$

(7) Go to step (2) unless all possible $\phi^{\mu\nu}_{max} \leq \epsilon$ with $\epsilon \ll 1$

(8) Set $\mathbf{Q} = \mathbf{Q}'$ and $\mathbf{u} = \mathbf{Q}\mathbf{y}$.

In the simulations presented below we will not use the ϵ criterion but simply set $\epsilon = 0$ and go through all possible pairs a fixed number of times in order to have a common criterion for all cumulant based methods (see below).

We refer to this algorithm as CuBICA (**Cu**mulant **B**ased **I**ndependent **C**omponent **A**nalysis) and indicate the different variants by appending the order of cumulants used in the contrast. For example a variant with contrast function based on 3rd and 4th order information is called CuBICA34. approximate contrast functions (see below) are indicated by an additional 'a', e.g., CuBICA34a.

E. Convergence

Since Ψ_{34} is a contrast it has the property

$$\Psi_{34}(\mathbf{M}\mathbf{u}) \leq \Psi_{34}(\mathbf{u}) \quad \forall\, \mathbf{M} \text{ orthogonal}, \quad (14)$$

if $\mathbf{u}$ has maximally independent components [7]. In the algorithm one can divide $\Psi_{34}(\mathbf{Q}^{\mu\nu}, \mathbf{y}')$ in (7) for every new Givens rotation $\mathbf{Q}^{\mu\nu}$ into two parts. One part is not affected by the rotation and the other is $\Psi^{\mu\nu}_{34}(\phi, \mathbf{y}')$. Since $\Psi^{\mu\nu}_{34}(\phi, \mathbf{y}')$ is maximized, $\Psi_{34}(\mathbf{Q}^{\mu\nu}, \mathbf{y}')$ and therefore also $\Psi_{34}(\mathbf{Q}', \mathbf{y})$ have to increase monotonically with every rotation. But $\Psi_{34}(\mathbf{Q}', \mathbf{y})$ has an upper bound, and thus will converge to a maximum. Of course we cannot rule out that there might be local maxima although they have not been observed.

F. Approximation of Ψ_{34}

Empirically we have found that the third term in (13) is small compared to the second one. In fact the amplitude of the third term, A_8, is about one magnitude smaller than that of the second term, A_4, independently of the chosen data sets (see Fig. 1). This suggests to neglect the third term and write as an approximate criterion

$$\tilde{\Psi}_{34}(\phi, \mathbf{y}) = A_0 + A_4 \cos\left(4\phi + \phi_4\right) . \qquad (15)$$

Note that $\tilde{\Psi}_{34}$ still takes third- *and* fourth-order cumulants into account. As in the exact case, unmixing criteria restricted to fourth-order cumulants, i.e. $\tilde{\Psi}_4$, can be easily obtained by setting all third-order cumulants to zero. The contrast function for third-order cumulants order only, $\tilde{\Psi}_3$ remains the same in the approximate form (15), since A_8 in (13) contains no information of third-order cumulants. Finding the maxima ϕ_{max} of (15) is trivial. They are the angles satisfying the condition

$$\phi_{max} = n\frac{\pi}{2} - \frac{\phi_4}{4}\,, \quad n\,\epsilon\,\{0, \pm1, \pm2, \pm3, \ldots\}\ . \qquad (16)$$

The maximum we chose is simply $\phi_{max} = -\frac{\phi_4}{4}$.

III. Comparison with Other Algorithms

We compared four variants of CuBICA, namely those based on $\Psi_{34}(\phi, \mathbf{y})$ (CuBICA34), $\tilde{\Psi}_{34}(\phi, \mathbf{y})$ (CuBICA34a), $\Psi_4(\phi, \mathbf{y})$ (CuBICA4), and $\tilde{\Psi}_4(\phi, \mathbf{y})$ (CuBICA4a), with Comon's original algorithm based on $\Psi_4(\xi, \mathbf{y})$ [1], with the JADE algorithm [15], which diagonalizes 4th order cumulant matrices, the Infomax Algorithm [16], and the FastICA package [6] using a fixed-point algorithm with different nonlinearities. In all cases we have used original software provided by the authors [1].

It is interesting to note that the different algorithms make different assumptions about the distributions of the sources. Infomax uses a one-parametric symmetrical model for the distributions of the sources and thus makes the assumptions very explicit. It is not clear to us which deviations of the true distribution from the model can degrade the unmixing performance and to what extent. Cumulant based methods, on the other hand, make no explicit assumptions about the source distributions. However, by focusing only on cumulants of low order and since cumulants of different order do not mix under a linear transformation, these methods are completely blind to higher order cumulants. Thus there is an implicit assumption that the distributions are such that low order cumulants contain enough information for the unmixing. Therefore considering fourth-order cumulants only is equivalent to a one-parametric model of the source distribution whereas considering third- and fourth-order cumulants results in a two-parametric family of functions. FastICA is similar in this respect using a one-parametric approach, although it is not restricted to cumulants but can also be derived using non-polynomial functions.

[1]Comon's algorithm: http://www.i3s.unice.fr/~comon/codesICA.txt (Version 6 of March 1992, downloaded December 12th, 2001); JADE: ftp://tsi.enst.fr/pub/jfc/Algo/Jade/jadeR.m (Version 1.5 of December 1997, downloaded March 6th, 2001); FastICA: http://www.cis.hut.fi/projects/ica/fastica/loadcode.shtml (Version 2.1 of January 15th, 2001, downloaded January 15th, 2001); Infomax: http://www.cnl.salk.edu/~tewon/ICA/Code/ext_ica_download.html (Version 2.0 of August 23rd, 1998, downloaded March 6th, 2001); CuBICA: http://itb.biologie.hu-berlin.de/~blaschke (Version 1.6 of February 22th, 2002)

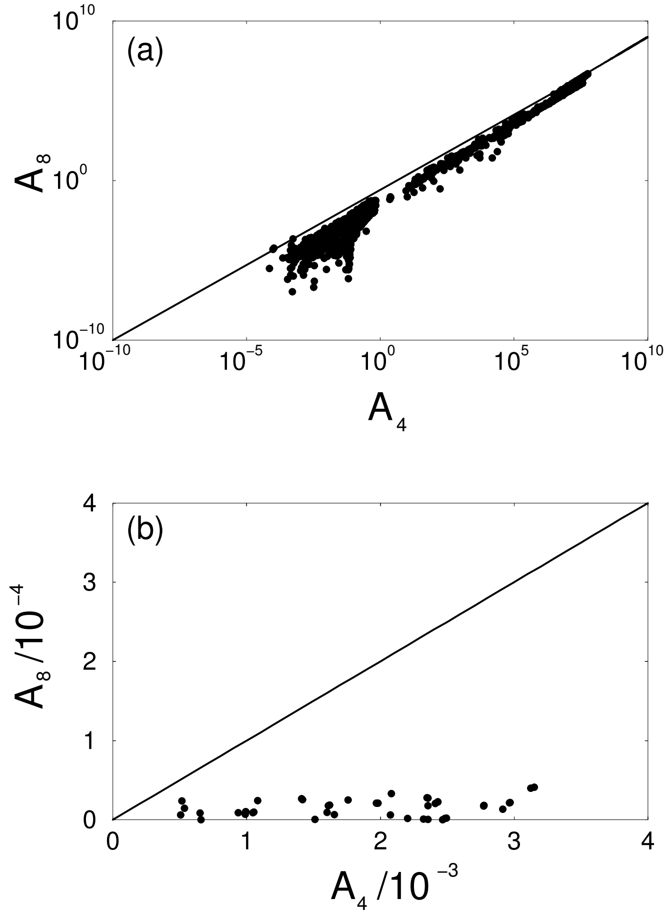

Fig. 1. Plot of amplitude A_8 versus A_4 for all rotations in two simulations with different data sets. The straight lines indicate $A_8 = 0.1 * A_4$. **(a)** Simulation with data set (v) (symmetrically distributed sources) and contrast function Ψ_4. Similar results were obtained by using Ψ_{34} as a contrast. Note the logarithmic axes. A_8 is about one magnitude smaller than A_4. Less than 1% of all values for $\frac{A_8}{A_4}$ exceed the 0.1-line. **(b)** Simulation with data set (ii) (non-symmetrically distributed sources) and contrast function Ψ_{34}. In this case the difference is even greater since for non-symmetrically distributed sources A_4 has additional terms from third order cumulants which do not appear in A_8.

A. Simulations

We assembled five different data sets of length 44218. Data set (i) contained real acoustic sources from [17] and [18]. Data set (ii) contained non-symmetrically distributed sources and (v) was composed of different symmetrically distributed sub- and super-Gaussian sources, both sets were generated synthetically. Set (iii) and (iv) were mixtures of real acoustic and synthetic sources. For further details see Table I. Each data set was mixed by a randomly chosen mixing matrix with entries chosen uniformly from $[-1, 1]$.

To simplify the comparison between the algorithms we used the same stopping criterion for all four cumulant based methods, namely we stopped after M sweeps through all possible pairs of signal components, where M is the nearest integer to $1+\sqrt{N}$ and N is the number of source components. Unmixing performance did not depend significantly on the stopping criterion, but comparison of the time performances is clearer with a common stopping criterion. Since it is not easy to define a similar criterion for FastICA and Infomax we did not change these algorithms.

To quantify the performances we slightly modified an error measure proposed by [19] and defined the unmixing error

$$E = \frac{1}{N^2}\left(\sum_{i=1}^{N}\left(\sum_{j=1}^{N}\frac{|P_{ij}|}{\max_k |P_{ik}|}-1\right)+\sum_{j=1}^{N}\left(\sum_{i=1}^{N}\frac{|P_{ij}|}{\max_k |P_{kj}|}-1\right)\right), \tag{17}$$

where $\mathbf{P}$ is the performance matrix $\mathbf{P} = \mathbf{QWA}$. Unmixing error E indicates good unmixing by low values and vanishes for perfect unmixing. An example of the development of E during a simulation is shown in Fig. 2.

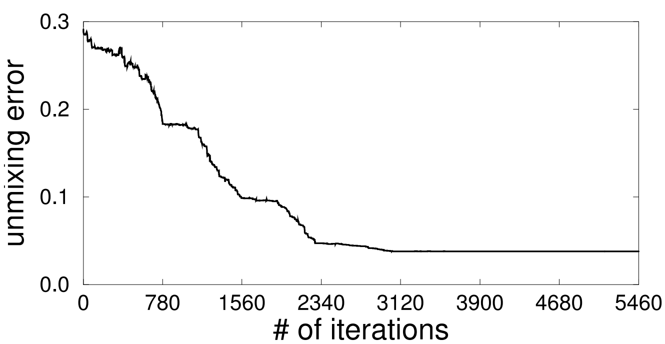

Fig. 2. Development of the unmixing error for data set (v) from Table I with $N = 40$ using Ψ_{34} as a contrast. The algorithm was stopped after 7 sweeps through all $N * (N-1)/2 = 780$ possible pairs of signal components.

Infomax and FastICA required some manual assistance, while the other algorithms could be applied directly. For Infomax we usually applied the algorithm to the data twice with different parameter settings. The first run did a rough unmixing which was then refined in the second run on the already roughly unmixed data. Several test runs were necessary to find appropriate parameter settings, which was quite time consuming. For FastICA we had to do test runs to determine the nonlinearity yielding best performance. In both cases the error criterion (17) guided the parameter selection, so that the results were not obtained completely unsupervised but with some supervision. Since the data sets were sufficiently long to rule out overfitting, we did not use separate training and test sets in these experiments.

To investigate the dependency of the algorithms on the length of the data set, we also did simulations on data set (v) with different numbers of data points and compared the unmixing errors (see Fig. 3). Data set (v) was split into 11 subsets of length T, with $T \in \{40, 80, 160, 320, 640, 1280, 2560, 5120, 10240, 20480\}$ (for $T = 5120$, 10240, and 20480, there were only 8, 4, and 2 pieces, respectively). The first subset was used to optimize the parameters of FastICA and Infomax for one given mixing matrix. Then all algorithms were tested with ten different mixing matrices on each of the remaining subsets.

B. Results

We measured unmixing errors and the elapsed time for all 8 different algorithms and 5 data sets of full length, see Table I. Since additional simulations with different mixing matrices showed no significant variations in the results, we only give here the mean values for unmixing error and time consumption.

For symmetrically distributed sources, all algorithms performed similarly well (data sets (i), (iv), (v)). If the sources were skew-symmetric, the additional third order information was crucial and CuBICA34, CuBICA34a, and FastICA using a corresponding nonlinearity clearly gave better results (data set (ii)). In case where the sources were both symmetric and skew-symmetric, only CuBICA34 and CuBICA34a with contrast function Ψ_{34} and $\tilde{\Psi}_{34}$, respectively, could discriminate between the different distributions (data set (iii)). Thus, the comparison in Table I suggests that different ICA-algorithms perform similarly well as long as their contrast functions are sensitive to the properties of the source distributions. Methods blind to skew-symmetric distributions fail on data set (ii) and only the methods that take third- *and* fourth-order cumulants into account can deal with mixtures of symmetric and skew-symmetric distributions (data set (iii)). The CuBICA-algorithms with approximate unmixing criterion gave similar unmixing errors as the algorithms using the exact contrast. This suggests that A_8 in Eq. (15) is indeed negligible. However, since there is no advantage over the algorithms using the exact contrast in terms of CPU-time, this is mainly of theoretical interest.

The complexity of Comon's algorithm and CuBICA is of the same order. A significant difference is the way how the optimal rotation angle $\phi^{\mu\nu}$ is found. Comon's algorithm uses a Matlab function to numerically find the root of a polynomial of degree four in each step whereas CuBICA generates an array of function values and searches for the maximal value. In Matlab implementation CuBICA performs faster and gives in general slightly smaller unmixing errors. JADE is an algorithm based on kurtosis-maximization. It uses a matrix-approximation for cumulant tensors of 4th order. This may explain the less accurate performance on data set (v) but is also responsible for the relatively high speed, since matrices can be processed efficiently in the Matlab implementation. This speed advantage might be less significant in a C implementation, for instance, and vanishes for larger N (see data set (v)) because JADE needs to compute all N^4 possible cumulants of 4th order at least once. Comon's algorithm and CuBICA, on the other hand, only need to compute cumulants with at most two different indices. FastICA is significantly faster and Infomax is much slower than the cumulant based methods. Both algorithms have in common that in our experiments they needed some manual assistance. In FastICA we had to decide which nonlinearity should be used. This decision was guided by the unmixing error, a measure that is usually not available in

TABLE I

UNMIXING ERROR (E) AND CPU-TIME IN SECONDS FOR DIFFERENT ALGORITHMS AND DATA SETS

contrast function/ algorithm	data sets, # of components (N) (i) N=6	(ii) N=6	(iii) N=7	(iv) N=12	(v) N=40	data sets, # of components (N) (i) N=6	(ii) N=6	(iii) N=7	(iv) N=12	(v) N=40
CuBICA34	0.017	0.039	0.041	0.038	0.039	1.5	1.4	2.8	10.1	**230.3**
CuBICA34a	0.018	0.040	0.042	0.038	0.038	1.4	1.4	2.7	9.8	**227.6**
CuBICA4	0.016	**0.31**	**0.11**	0.035	0.039	1.4	1.4	2.7	10.0	**222.8**
CuBICA4a	0.017	**0.32**	**0.12**	0.036	0.038	1.4	1.4	2.6	9.8	**218.8**
Comon	0.017	**0.25**	**0.14**	0.049	0.061	2.4	2.3	4.3	14.1	**300.2**
JADE	0.016	**0.30**	**0.11**	0.035	**0.10**	0.7	0.7	1.1	5.4	**404.6**
Infomax	0.018	**0.47**	**0.17**	0.043	0.035	**48.1**	**49.3**	**57.8**	**112.1**	**512.3**
FastICA	0.016	0.040	**0.11**	0.042	0.037	1.7	0.5	0.5	6.2	16.8

Data sets: (i) 5 real acoustic sources from [17] + 1 normally distributed source ($N(0,1)$), (ii) 5 skew-normally distributed sources [20] + 1 normally distributed source, (iii) 3 music sources from [18] + 3 skew-normally distributed sources + 1 normally distributed source, (iv) 6 real acoustic sources (3 speech a+ 3 music sources) from [17] and [18] + 3 Laplace distributed sources + 1 normally distributed source + 1 skew-normally distributed source + 1 $\sin(0.05 * t)$, (v) 10 Beta distributed sources (super-Gaussian) + 10 Cauchy distributed sources (sub-Gaussian) + 10 Laplace distributed sources (super-Gaussian) + 10 Student-t distributed sources (sub-Gaussian). The number of data points for all data sets was T=44218. Additional 20 simulations with different mixing matrices showed no significant variations in the unmixing errors. Low values of E indicate good performance. Times have been measured on a 1.8 GHz Pentium IV PC using Matlab 6.0 implementation. Relatively large unmixing errors and long CPU-times are set bold face.

more realistic applications. Infomax required some parameter tuning and repeated application to the data with different parameter sets which made the algorithm inconvenient to use. It also seems questionable whether the speed advantages of JADE and FastICA are worth the worse performance and the required manual assistance, respectively.

By comparing unmixing errors of the different algorithms depending on the number of data points N one can see from Fig. 3 that all methods degrade similarly with shortened length of the data sets. One marked difference however was that Infomax had a large variance, while all other algorithms gave virtually identical results over different simulation runs. Thus although Infomax yielded best performance in some runs it performed worst in others and we found it to be unreliable, particularly on the short data sets. On the long data sets used for Table I Infomax was nearly as reliable as the other algorithms.

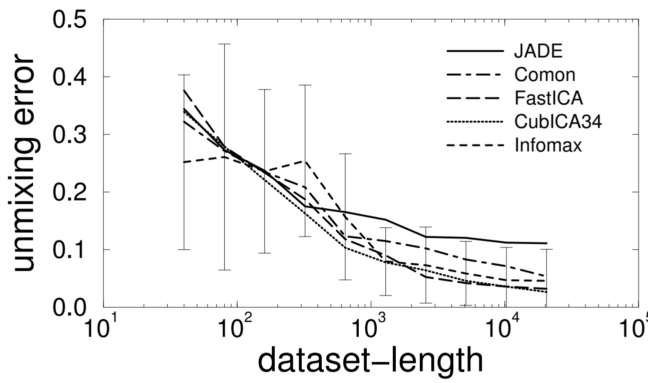

Fig. 3. Mean unmixing errors for data set (v) from Table I with $N = 40$ components and different numbers of data points T with $T \in \{40, 80, 160, 320, 640, 1280, 2560, 5120, 10240, 20480\}$. For each T we took 10 different samples and performed 10 simulations on each, every simulation with a different mixing matrix. For $T = 5120, 10240$, and 20480 we used $7, 3$, and 1 different samples, respectively, due to the length of the whole data set (v). The standard deviation of the unmixing errors was less then 0.01 for all algorithms except Infomax. For Infomax the parameter set with smallest unmixing error was found on a training set mixed with a single mixing matrix. We used only one mixing matrix because finding a good parameter set for several mixing matrices was too time consuming since the algorithm did often not converge on one of the matrices. Results shown here are for the same test data sets as for the other algorithms. The errorbars denote twice the standard deviation of the unmixing error of Infomax.

IV. CONCLUSION

We have proposed CuBICA, an improved cumulant based method for independent component analysis. In contrast to Comon's method [1] and other algorithms (FastICA, Infomax, and JADE) it takes third- and fourth-order statistics into account simultaneously (CuBICA34) and is thus able to handle linear mixtures of symmetrically and skew-symmetrically distributed source signal components. Due to its mathematically simple formulation and since A_8 in Eq. (15) is small compared to A_4, approximate algorithms, CuBICA34a and CuBICA4, can be derived easily, which show equal performances. Although, this is mainly of theoretical interest, since the approximate algorithms are not significantly faster. Furthermore, in contrast to FastICA and Infomax, CuBICA can be used without any parameter adjustments.

Since CuBICA can handle symmetric and asymmetric distributed sources, is easy to use, and shows good performance, it may be a good general algorithm for performing ICA.

APPENDIX

A. Constants in Equation (10)

The definitions of d_{ni} follow directly from the multilinearity of $C^{(\mathbf{u})}_{\cdots}$:

$$d_{30} := \left({C^{(\mathbf{y})}_{111}}^2 + {C^{(\mathbf{y})}_{222}}^2 \right) \quad ,$$

$$d_{31} := 6 \left(C^{(\mathbf{y})}_{111} C^{(\mathbf{y})}_{112} - C^{(\mathbf{y})}_{122} C^{(\mathbf{y})}_{222} \right) \quad ,$$

$$d_{32}{:=}9\left({C^{(\mathbf{y})}_{112}}^2+{C^{(\mathbf{y})}_{122}}^2\right)+6\left(C^{(\mathbf{y})}_{111}C^{(\mathbf{y})}_{122}+C^{(\mathbf{y})}_{112}C^{(\mathbf{y})}_{222}\right) \ ,$$
$$d_{33}{:=}2\,C^{(\mathbf{y})}_{111}C^{(\mathbf{y})}_{222}+18\,C^{(\mathbf{y})}_{112}C^{(\mathbf{y})}_{122} \ ,$$
$$d_{40}{:=}\left({C^{(\mathbf{y})}_{1111}}^2+{C^{(\mathbf{y})}_{2222}}^2\right) \ ,$$
$$d_{41}{:=}8\left(C^{(\mathbf{y})}_{1111}C^{(\mathbf{y})}_{1112}-C^{(\mathbf{y})}_{1222}C^{(\mathbf{y})}_{2222}\right) \ ,$$
$$d_{42}{:=}16\left({C^{(\mathbf{y})}_{1112}}^2+{C^{(\mathbf{y})}_{1222}}^2\right)+ 12\left(C^{(\mathbf{y})}_{1111}C^{(\mathbf{y})}_{1122}+C^{(\mathbf{y})}_{1122}C^{(\mathbf{y})}_{2222}\right) \ ,$$
$$d_{43}{:=}48\left(C^{(\mathbf{y})}_{1112}C^{(\mathbf{y})}_{1122}-C^{(\mathbf{y})}_{1122}C^{(\mathbf{y})}_{1222}\right) +8\left(C^{(\mathbf{y})}_{1111}C^{(\mathbf{y})}_{1222}-C^{(\mathbf{y})}_{1112}C^{(\mathbf{y})}_{2222}\right) \ ,$$
$$d_{44}{:=}36\,{C^{(\mathbf{y})}_{1122}}^2+32\,C^{(\mathbf{y})}_{1112}C^{(\mathbf{y})}_{1222}+2\,C^{(\mathbf{y})}_{1111}C^{(\mathbf{y})}_{2222} \ .$$

B. Constants in Equation (13)

From (10) one can derive

$$\Psi_n(\phi,\mathbf{y}){=}a_{n0}+s_{n4}\sin(4\phi)+c_{n4}\cos(4\phi) +s_{n8}\sin(8\phi)+c_{n8}\cos(8\phi) \quad \text{for } n\in\{3,4\}\,,$$

with

$$a_{30}{:=}\frac{1}{3!}\frac{1}{8}\left[5\left({C^{(\mathbf{y})}_{111}}^2+{C^{(\mathbf{y})}_{222}}^2\right) + 9\left({C^{(\mathbf{y})}_{112}}^2+{C^{(\mathbf{y})}_{122}}^2\right)+6\left(C^{(\mathbf{y})}_{111}C^{(\mathbf{y})}_{122}+C^{(\mathbf{y})}_{112}C^{(\mathbf{y})}_{222}\right)\right] \ ,$$
$$a_{40}{:=}\frac{1}{4!}\frac{1}{64}\left[35\left({C^{(\mathbf{y})}_{1111}}^2+{C^{(\mathbf{y})}_{2222}}^2\right)+\ 80\left({C^{(\mathbf{y})}_{1112}}^2+{C^{(\mathbf{y})}_{1222}}^2\right) +60\left(C^{(\mathbf{y})}_{1111}C^{(\mathbf{y})}_{1122}+C^{(\mathbf{y})}_{1122}C^{(\mathbf{y})}_{2222}\right) + 108\,{C^{(\mathbf{y})}_{1122}}^2+96\,C^{(\mathbf{y})}_{1112}C^{(\mathbf{y})}_{1222}+6\,C^{(\mathbf{y})}_{1111}C^{(\mathbf{y})}_{2222}\right] \ ,$$
$$s_{34}{:=}\frac{1}{3!}\frac{1}{4}\left[6\left(C^{(\mathbf{y})}_{111}C^{(\mathbf{y})}_{112}-C^{(\mathbf{y})}_{122}C^{(\mathbf{y})}_{222}\right)\right] \ ,$$
$$c_{34}{:=}\frac{1}{3!}\frac{1}{8}\left[3\left({C^{(\mathbf{y})}_{111}}^2+{C^{(\mathbf{y})}_{222}}^2\right) - 9\left({C^{(\mathbf{y})}_{112}}^2+{C^{(\mathbf{y})}_{122}}^2\right)-6\left(C^{(\mathbf{y})}_{111}C^{(\mathbf{y})}_{122}+C^{(\mathbf{y})}_{112}C^{(\mathbf{y})}_{222}\right)\right] \ ,$$
$$s_{44}{:=}\frac{1}{4!}\frac{1}{32}\left[56\left(C^{(\mathbf{y})}_{1111}C^{(\mathbf{y})}_{1112}-C^{(\mathbf{y})}_{1222}C^{(\mathbf{y})}_{2222}\right) + 48\left(C^{(\mathbf{y})}_{1112}C^{(\mathbf{y})}_{1122}-C^{(\mathbf{y})}_{1122}C^{(\mathbf{y})}_{1222}\right) +8\left(C^{(\mathbf{y})}_{1111}C^{(\mathbf{y})}_{1222}-C^{(\mathbf{y})}_{1112}C^{(\mathbf{y})}_{2222}\right)\right] \ ,$$
$$c_{44}{:=}\frac{1}{4!}\frac{1}{16}\left[7\left({C^{(\mathbf{y})}_{1111}}^2+{C^{(\mathbf{y})}_{2222}}^2\right)-\ 16\left({C^{(\mathbf{y})}_{1112}}^2+{C^{(\mathbf{y})}_{1222}}^2\right) -12\left(C^{(\mathbf{y})}_{1111}C^{(\mathbf{y})}_{1122}+C^{(\mathbf{y})}_{1122}C^{(\mathbf{y})}_{2222}\right) - 36\,{C^{(\mathbf{y})}_{1122}}^2-32\,C^{(\mathbf{y})}_{1112}C^{(\mathbf{y})}_{1222}-2\,C^{(\mathbf{y})}_{1111}C^{(\mathbf{y})}_{2222}\right] \ ,$$
$$s_{38}{:=}0 \ ,$$
$$c_{38}{:=}0 \ ,$$
$$s_{48}{:=}\frac{1}{4!}\frac{1}{64}\left[8\left(C^{(\mathbf{y})}_{1111}C^{(\mathbf{y})}_{1112}-C^{(\mathbf{y})}_{1222}C^{(\mathbf{y})}_{2222}\right) -48\left(C^{(\mathbf{y})}_{1112}C^{(\mathbf{y})}_{1122}-C^{(\mathbf{y})}_{1122}C^{(\mathbf{y})}_{1222}\right) -8\left(C^{(\mathbf{y})}_{1111}C^{(\mathbf{y})}_{1222}-C^{(\mathbf{y})}_{1112}C^{(\mathbf{y})}_{2222}\right)\right] \ ,$$
$$c_{48}{:=}\frac{1}{4!}\frac{1}{64}\left[\left({C^{(\mathbf{y})}_{1111}}^2+{C^{(\mathbf{y})}_{2222}}^2\right)-16\left({C^{(\mathbf{y})}_{1112}}^2+{C^{(\mathbf{y})}_{1222}}^2\right) -12\left(C^{(\mathbf{y})}_{1111}C^{(\mathbf{y})}_{1122}+C^{(\mathbf{y})}_{1122}C^{(\mathbf{y})}_{2222}\right) + 36\,{C^{(\mathbf{y})}_{1122}}^2+32\,C^{(\mathbf{y})}_{1112}C^{(\mathbf{y})}_{1222}+2\,C^{(\mathbf{y})}_{1111}C^{(\mathbf{y})}_{2222}\right] \ .$$

With this it is trivial to determine the constants for $\Psi_{34}(\phi,\mathbf{y})=\Psi_3(\phi,\mathbf{y})+\Psi_4(\phi,\mathbf{y})$ in the form given in (13). We find:

$$A_0{:=}a_{30}+a_{40} \ ,$$
$$A_4{:=}\sqrt{(c_{34}+c_{44})^2+(s_{34}+s_{44})^2} \ ,$$
$$A_8{:=}\sqrt{c_{48}^2+s_{48}^2} \ ,$$
$$\tan(\phi_4){:=}-\frac{s_{34}+s_{44}}{c_{34}+c_{44}} \ ,$$
$$\tan(\phi_8){:=}-\frac{s_{48}}{c_{48}} \ .$$

The coefficients A_0, A_4 and ϕ_4 are functions of the cumulants of 3rd and 4th order of the centered and whitened signal $\mathbf{y}$. A_8 and ϕ_8 depend only on the 4th order cumulants.

ACKNOWLEDGMENT

We thank Raphael Ritz for helpful comments on the manuscript.

REFERENCES

[1] P. Comon, "Tensor diagonalization, a useful tool in signal processing," in *IFAC-SYSID, 10th IFAC Symposium on System Identification*, M. Blanke and T. Soderstrom, Eds., vol. 1, Copenhagen, Denmark, 1994, pp. 77–82.

[2] T.-W. Lee, M. Girolami, A. Bell, and T. Sejnowski, "A unifying framework for independent component analysis," *International Journal of Computers and Mathematics with Applications*, vol. 39, no. 11, pp. 2–21, 2000.

[3] A. Hyvärinen, J. Karhunen, and E. Oja, *Independent Component Analysis*, ser. Wiley Series on Adaptive and Learning Systems for Signal Processing, Communications, and Control. New York: John Wiley&Sons, 2001.

[4] A. Cichocki and S. Amari, *Adaptive Blind Signal and Image Processing*. New York: John Wiley&Sons, 2002.

[5] A. Bell and T. Sejnowski, "An information-maximisation approach to blind separation and blind deconvolution," *Neural Computation*, vol. 6, no. 7, pp. 1004–1034, 1995.

[6] A. Hyvärinen, "Fast and robust fixed-point algorithms for independent component analysis," *IEEE Transactions on Neural Networks*, vol. 10, no. 3, pp. 626–634, 1999.

[7] P. Comon, "Independent Component Analysis, a new concept?" *Signal Processing, Elsevier*, vol. 36, no. 3, pp. 287–314, 1994, special issue on Higher-Order Statistics.

[8] J.-F. Cardoso, "High-order contrasts for independent component analysis," *Neural Computation*, vol. 11, no. 1, pp. 157–192, 1999.

[9] A. Hyvärinen, "Survey on independent component analysis," *Neural Computing Surveys*, vol. 2, pp. 94–128, 1999.

[10] P. McCullagh, *Tensor Methods in Statistics*, ser. Monographs on Statistics and Applied Probability. London: Chapmann and Hall, 1987.

[11] G. Deco and D. Obradovic, *An Information-Theoretic Approach to Neural Computing*, ser. Springer Series in Perspectives in Neural Computing. New York: Springer, 1996.

[12] P. Comon, "Independent component analysis, contrasts and convolutive mixtures," in *Second IMA Conference on Mathematics in Communications*, Lancaster, UK, 2002.

[13] E. Moreau and N. Thirion-Moreau, "Nonsymmetrical contrasts for sources separation," *IEEE Transactions on Signal Processing*, vol. 47, no. 8, pp. 2241–2252, 1999.

[14] L. De Lathauwer, B. De Moor, and J. Vandewalle, "Blind source separation by simultaneous third-order tensor diagonalization," in *Proc. of the 8th European Signal Processing Conference (EUSIPCO'96)*, Trieste, Italy, 1996, pp. 2089–2092.

[15] J.-F. Cardoso and A. Souloumiac, “Blind beamforming for non Gaussian signals,” *IEE Proceedings-F*, vol. 140, pp. 362–370, 1993.

[16] T.-W. Lee, M. Girolami, and T. Sejnowski, “Independent component analysis using an extended infomax algorithm for mixed sub-gaussian and super-gaussian sources,” *Neural Computation*, vol. 11, no. 2, pp. 409–433, 1999.

[17] John Fitzgerald Kennedy Library, Boston, “Sound excerpts from the speeches of president John F. Kennedy,” http://www.jfklibrary.org/speeches.htm, 1996, retrieved February 6th, 2002.

[18] B. Pearlmutter, “16 clips sampled from audio CDs,” http://snot.cs.unm.edu/˜bap/cICA/clips-wav/, 1996, retrieved February 7th, 2002.

[19] S. Amari, A. Cichocki, and H. Yang, “Recurrent neural networks for blind separation of sources,” in *Proc. of the Int. Symposium on Nonlinear Theory and its Applications (NOLTA-95)*, Las Vegas, USA, 1995, pp. 37–42.

[20] A. Azzalini, “A class of distributions which includes the normal ones,” *Scandinavian Journal of Statistics*, vol. 12, pp. 171–178, 1985.

PLACE
PHOTO
HERE

Tobias Blaschke received the diploma degree in theoretical physics from the Free University, Berlin, Germany, in 1999.

He is currently working toward the Ph.D. degree under the supervision of Laurenz Wiskott at the Institute for Theoretical Biology, Humboldt University, Berlin. His research interests include statistical signal processing with emphasis on blind source separation and independent component analysis.

Laurenz Wiskott studied physics in Göttingen and Osnabrück, did his Ph.D. in 1995 in the group of Christoph von der Malsburg at the Ruhr-University Bochum, Germany, and then worked in the group of Terrence Sejnowski at the Salk Institute for Biological Studies, San Diego. After a one year fellowship at the Institute for Advanced Studies in Berlin he joined the group of Andreas V. M. Herz at the Humboldt-University Berlin August 1999. Since November 2000 he has a junior research group working on self-organization and unsupervised learning in the visual system including algorithms for independent component analysis.