

A FAST MULTILINEAR ICA ALGORITHM

Raghu G. Raj
U.S. Naval Research Laboratory
Washington D.C. 20375

Alan C. Bovik
The University of Texas at Austin
Austin, TX 78712

ABSTRACT

We extend our previous work on Multilinear Independent Component Analysis (MICA) by introducing a Fast-MICA algorithm that demonstrates the same improvement over classical ICA as the original MICA algorithm [1] while improving the computational speed by two polynomial orders of magnitude. Apart from enabling a faster determination of the multilinear structure of image patch probability density, this new approach opens up, for the first time, the possibility of computing a novel non-stationarity index based on the relative change in mutual information. We demonstrate the performance of our Fast-MICA algorithm together with an illustration of our novel non-stationarity index.

Index Terms—ICA, non-stationarity, fixation selection

1. INTRODUCTION

Denote the probability of the source that we are modeling by $P(X)$, where X is a random vector whose realizations have dimensionality d . The goal of ICA is to factor the probability distribution of the source into a product of distributions: $P(X) = \prod_{i=1}^d p(z_i)$, where $\{z_i = X * \phi_i\}_{i=1}^d$ are

filtered responses of the source. The filters $\{\phi_i\}_{i=1}^d$ are the ICA filters of the source. Statistical algorithms for computing the ICA filters have been the subject of intense study [2], most of which involve the construction of different cost functions (usually variations of the maximum likelihood cost function).

The independent directions that emerge from an ICA decomposition can be fruitfully utilized by reducing the d -dimensional problem into d independent 1-D problems. Furthermore, ICA decompositions of data having heavy tailed marginals (as is for example observed in NSS applications) tend to favor sparse representations. Sparse representations are useful for many applications that seek to efficiently represent and process the data.

However, in spite of these potential advantages, in reality the statistics of most real-world sources, such as natural image patches, cannot be strictly factored into a simple product. As a result, the so-called independent components contain significant mutual dependencies between them [2]. Accordingly, prior work has attempted to more completely capture statistical image structure by accounting for the

dependencies (either directly or indirectly) between the ICA components [3].

In [1] we approached this problem from the perspective of refining the classical ICA model such that the dependencies between pseudo-independent components are captured using a multilinear representation of $P(X)$:

$$P(X) = \frac{1}{Z} g(J) \prod_{i=1}^d p(z_i),$$

where $g: J = [z_1, \dots, z_d] \rightarrow R$ and $Z \in R$ is a normalizing constant. We call the resulting model the *Multilinear ICA (MICA) decomposition* of the distribution $P(X)$. Of all possible multilinear expansions of this form that could describe the source distribution, we seek the one that makes the representation of the source as sparse as possible, i.e., which minimizes the contribution of $g(J)$.

In [1] we successfully deployed the new method to model natural scene textures and demonstrated advantages relative to classical ICA. In Section 2 of this paper we first review the basic MICA model and also describe its computational complexity. Then in Section 3 we proceed to describe the construction of a Fast-MICA algorithm that enables the construction of the MICA distribution in $O(nd^2)$ (which is an improvement over the $O(nd^d)$ complexity of the original MICA algorithm [1]). Section 4 describes a novel application of the MICA for the computation of a mutual information based non-stationarity index for natural images. The computation of this index—which has been numerically infeasible until now—is facilitated by the Fast-MICA algorithm. Finally in Section 5 we show simulation results of the performance of the Fast-MICA algorithm along with the mutual information based non-stationarity map.

2. MULTILINEAR ICA

Consider the classical ICA model where the observation vector is modeled: $x = Bz$; where $x = [x_1, \dots, x_d]^T \in R^d$, $z = [z_1, \dots, z_d]^T \in R^d$, d is the intrinsic dimensionality of the data, and $B \in R^{d \times d}$ is a full-rank matrix. The goal of ICA is to find a matrix B such that the resulting components of z are independent random variables.

However, for many real-world sources, such as natural images, such an ideal decomposition is not possible and so the components of z will contain residual dependencies. Our aim is to explicitly capture these dependencies. In doing so we must first recognize that z cannot be further decomposed as a combination of independent sources via another full-

rank matrix! It is possible, however, that z can be decomposed with respect to an under-complete linear model, but this requires knowledge of the subspace dimensionality.

An alternate view which we explored in [1] is that, given knowledge of the intrinsic dimensionality d , the residual dependencies can be captured via non-linear combinations of independent sources. The choice of the non-linearity, as well as of the source distribution, must be as simple as possible, and yet must successfully account for the probabilistic structure of the observed natural image sources.

Perhaps the simplest non-linear system that one can hypothesize for natural image source modeling is a quadratic channel. In our experiments with natural image textures, we found that the hybrid linear-quadratic model (stimulated by *i.i.d.* Gaussian sources) shown in Fig. 1 can successfully account for the probabilistic structure of natural image patches. We now describe this non-linear system some detail.

The observable image source data that we are modeling is $x \in R^d$. $B \in R^{d \times d}$ is a full-rank matrix initially chosen as the matrix associated with the classical ICA decomposition of x which will be re-estimated in subsequent iterations. The system F in Fig. 1 models the residual interaction between the components of $z \in R^d$. It consists of a core non-linearity φ preceded by a linear system $y = As + \gamma$, where $y = [y_1, \dots, y_d]^T \in R^d$, $\gamma = [\gamma_1, \dots, \gamma_d]^T \in R^d$, and $s = [s_1, \dots, s_d]^T \in R^d$ are *i.i.d.* Gaussian: $s_i \sim \mathcal{N}(0,1)$. The density of the i^{th} Gaussian channel is denoted $q(s_i)$. The Gaussian channel variances are $\sigma = [\sigma_1, \dots, \sigma_d]^T \in R^d$, $\mu = [\mu_1, \dots, \mu_d]^T \in R^d$ is an additive mean adjusting vector, and $\beta = [\beta_1, \dots, \beta_d] \in R^d$ is a multiplicative vector that is applied (component-wise) to all channels, and which determines the effective non-linearity of the channels. Finally, $C = [C_{i,j}] = [C_1^T, \dots, C_d^T]^T = A^{-1} \in R^{d \times d}$ is an invertible linear transformation of the *i.i.d.* Gaussian sources that determines the interaction of the Gaussian sources.

Given this it was demonstrated in [1] that under the MICA distribution can be expressed in closed form as follows:

$$p(z) = \frac{K}{|J(F)|} g(J) \prod_{k=1}^d p((z_k - \mu_k)\beta_k) \quad (1)$$

where $p[(z_k - \mu_k)\beta_k] = K_i \exp\left(-a_i \left[\tilde{\varphi}[\beta_i(z_k - \mu_k)] - c_i\right]^2\right)$, $\tilde{\varphi} = \varphi^{-1}$,

and where K and $|J(F)|$ have closed form expressions with respect to the MICA parameters $(C, \sigma, \gamma, \mu, \beta)$. Equation (1) can be used to derive gradient equations for each of the parameters [1].

The computational complexity of MICA is $O(nd^4)$ primarily due to the following gradient equation required to update the matrix C [1]:

$$\frac{\partial \log[p(z)]}{\partial C_{m,n}} = \frac{(-1)^{m+n} |C^{m,n}| \operatorname{sgn}|C|}{\operatorname{abs}(|C|)} \left(\frac{C_m^T \gamma}{\sigma_m^2} \right) \gamma_n - \frac{C_{m,n}}{\sigma_m^2} \{\tilde{\varphi}[\beta_n(z_n - \mu_n)]\}^2 + \tilde{\varphi}[\beta_n(z_n - \mu_n)] \left[\frac{C_m^T \gamma}{\sigma_m^2} + \frac{C_{m,n} \gamma_n}{\sigma_m^2} \right] - \sum_{i \neq n} \frac{C_{m,i}}{\sigma_m^2} \tilde{\varphi}[\beta_i(z_i - \mu_i)] \tilde{\varphi}[\beta_n(z_n - \mu_n)] \quad (2)$$

where n is the number of samples and d is the dimension of the vector.

3. FAST-MICA ALGORITHM

Thus far our approach has been to derive gradient equations for each of the parameters based on the log likelihood equation derived from the MICA distribution in conjunction with heuristics for assessing the β parameter. We note that the central computational bottleneck in determining the MICA parameters by this method lies in the computation of the matrix C .

Figure 2 presents an alternative view of the computational structure involved in the training of the MICA model (as compared to Figure 1 which showed the same from generative point of view). Let us assume that the matrix B (either ICA or Gabor) along with parameters β and μ have already been estimated (using the initialization and heuristics used in the MICA paper). Thus what remains to be estimated are matrix C and parameter γ . We have the following relationship: $s = C(y - \gamma)$, such that we want to make the components of vector s *i.i.d.* Gaussian. This is equivalent to performing a PCA decomposition of y since, under the assumption that y is jointly Gaussian (which is a consequence of the requirement that s is jointly Gaussian), a PCA decomposition of y will yield uncorrelated Gaussian components which are consequently *i.i.d.* Let vectors $\{y_i\}_{i=1}^N$ be the N observation vectors (where $y_i \in R^d$); then our goal is to find the optimal parameter γ and optimal orthogonal matrix C such that we obtain a least-squares reconstruction error: $\min_{C, \gamma} \sum_{i=1}^N \|y_i - \gamma - C^T s\|^2$.

Firstly, it can be deduced that at the optimum γ will be $\gamma_{opt} = \bar{y} = \frac{1}{N} \sum_{i=1}^N y_i$. Given this, the optimal matrix C can be deduced by the standard procedure of eigenvalue decomposition of the correlation matrix of $(y - \gamma)$. In the simulations below we use a standard PCA algorithm to perform this decomposition which reduces the computational complexity to $O(nd^2)$. Alternatively one could also employ efficient Neural Network implementations of PCA such as the APEX algorithm which allows one to perform adaptive updates on the neural network on a sample-by-sample basis to yield a PCA decomposition.

4. NON-STATIONARITY INDEX

We now demonstrate how the Fast-MICA algorithm finds an important application in computing a non-stationarity index based on relative change in mutual information.

Fundamentally non-stationarity in an image is the comparison of the statistical properties of two different regions of the image. Though KLD (Kullback Leibler Divergence) is a natural way to quantify the non-stationarity between two regions in an image, it is impractical due to the difficulties in computing sufficiently rich density functions (owing to the high-dimensional structure of the data), compounded with the subsequent difficulty of numerically computing the KLD integral. In our previous work we have found it convenient to study the non-stationary structure of natural images via the so-called NANS (Natural Image Non-stationarity) index [4] which we defined as follows:

Definition 1 (NANS index): Let $\phi_x \equiv \{\phi_i\}_i$ be the set of MICA filters associated with a windowed region R_x about location x in an image. Then we define the NANS (Natural Image Non-stationarity) index at x as: $NANS_x =$

$$\left| \frac{d \ln(MSE_{x,\phi_x})}{dx} \right|; \text{ where, } MSE_{x,\phi_x} \text{ is the mean-squared error associated with the region } R_x \text{ and MICA filters } \phi_x: \\ MSE_{x,\phi_x} = \min_{\{c_k\}_k} E \left[\left(J - \sum_k c_k \phi_k \right)^2 \right] \quad \square$$

In similar fashion, it is natural to define a non-stationarity index based on the relative change of mutual information:

Definition 2 (MI index): Let $\phi_x \equiv \{\phi_i\}_i$ be the set of MICA filters associated with a windowed region R_x about location x in an image. Then we define the mutual information (MI) based non-stationarity index at x as: $MI_x =$

$$\left| \frac{d \ln(I_{x,\phi_x})}{dx} \right|; \text{ where, } I_{x,\phi_x} \text{ is the mutual information associated with the region } R_x \text{ and MICA filters } \phi_x:$$

$$I_{x,\phi_x} = D(p(J_1, \dots, J_d) || \prod_i p(J_i))$$

where, $J_i \equiv$ filtered image patch random variable with respect to MICA filter ϕ_i . \square

The discrete versions of the above indices are respectively $\left| \frac{MSE_{c,\phi_x} - MSE_{s,\phi_x}}{MSE_{c,\phi_x}} \right|$ and $\left| \frac{I_{c,\phi_x} - I_{s,\phi_x}}{I_{c,\phi_x}} \right|$; where I_{x,ϕ_x} is the mutual information and MSE_{x,ϕ_x} is the mean-squared error associated with the region R_x and MICA filters ϕ_x .

The computation of the MI index is numerically feasible now due to the Fast-MICA algorithm.

5. SIMULATION RESULTS

We define the $M \times M$ image patch statistics of an $N \times N$ image region to be the joint distribution of the random variables (pixel values) from $M \times M$ patches that sample the image region. In this paper we are specifically interested in modeling the $M \times M$ image patch statistics of natural scenes. In our simulations we chose $M=3$.

We uniformly sampled texture images obtained from the USC-SIPI Brodatz database [5] with $N_{patch} = 2000$ patches of size $M \times M$. An ICA was then performed on the data vectors obtained from each texture using Comon's algorithm [2] to obtain the matrix B . Subsequently the parameters $(C, \sigma, \gamma, \mu, \beta)$ of the MICA model were estimated as described in Section 2. The parameter β , as mentioned

earlier, was estimated heuristically using the criteria developed in [1] at the outset of the simulation and held to a constant value throughout. Parameter initialization procedures consistent with that described in [1] was performed.

Thereafter, for each texture, we compared the data distribution of each channel derived from test data sets (different from the training data sets) to the corresponding distribution predicted by the ICA and MICA models. In addition, the average of all the data channels was also compared with that predicted by the ICA and MICA models. Simulation of the MICA model was accomplished by generating d *i.i.d.* zero mean, unit-variance Gaussian channels as shown in Fig. 1, and plotting the histograms outputs of the channels when the optimal parameters (for the texture being modeled) were used.

Figures 3 and 4 depict texture images taken from the Brodatz database [4]. Figs. 3(a)-3(f) and 4(a)-4(f) show the histograms of two of the channels corresponding to each of the textures, as well as both the corresponding computed ICA distributions and the corresponding computed MICA distributions. Also shown in Figs. 3(g-i) and 4(g-i) are the histograms of the data distributions when all of the data channels of the corresponding textures are averaged together as well as the corresponding computed ICA and MICA distributions. From the results it is qualitatively and quantitatively established that the MICA model allows for significantly improved approximation of the original data distributions as compared to the classical ICA model.

Figure 5(a) shows a grass-water multi-texture image and Figures 5(b) and 5(c) show respectively the NANS and MI non-stationarity maps using a *center-surround (CS) architecture* [4]. Firstly the use of CS architecture implies that there should be *two lines* present in the map on either side of the texture transition; this however absent in the NANS case due to its lack of symmetrical processing of the textures. In spite of this, for this example, the NANS index does appear to furnish visually better results than the MI map since the central line extends through the entire texture transition region. Nevertheless, the quantitative performance of the MI index for *human fixation section tasks* and its comparison to the NANS index [4] is an important unanswered question which is a subject of our future work.

REFERENCES

- [1] R.G. Raj and A.C. Bovik, "MICA: A multilinear ICA decomposition for natural image modeling," *IEEE Transactions on Image Processing*, vol. 17, no: 3, March 2008, Page(s): 259-271.
- [2] P. Comon, "Independent component analysis: A new concept?," *Signal Process.*, vol. 36, no. 3, pp. 287-314, Apr. 1994.
- [3] A. Hyvarinen, P.O. Hoyer, and M. Inki, "Topographic independent component analysis," *Neural Comput.*, vol. 13, no. 7, pp. 1527-1558, July 2001.
- [4] R.G. Raj, A.C. Bovik and L.K. Cormack, "Fixture selection by maximization of texture and contrast information," *IEEE Int. Conf. on Image Processing*, San Diego, California, October 12-15, 2008.
- [5] <http://sipi.usc.edu/database/database.cgi?volume=textures>

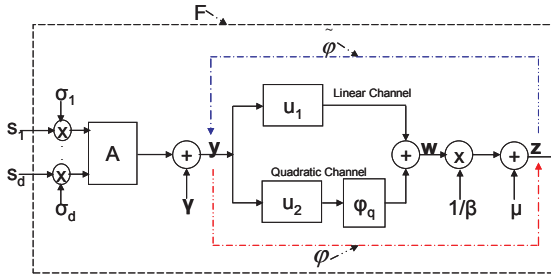


Fig.1: Generative non-linear model of MICA distribution

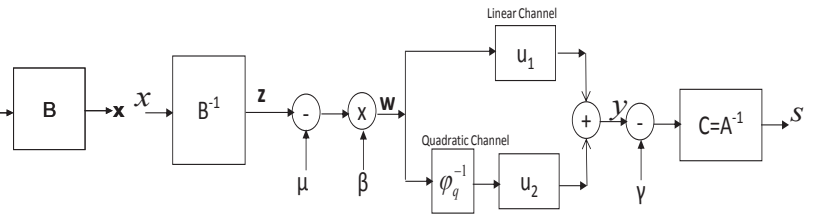


Fig.2: Training model of the MICA distribution

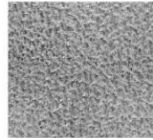


Fig 3: Pigskin

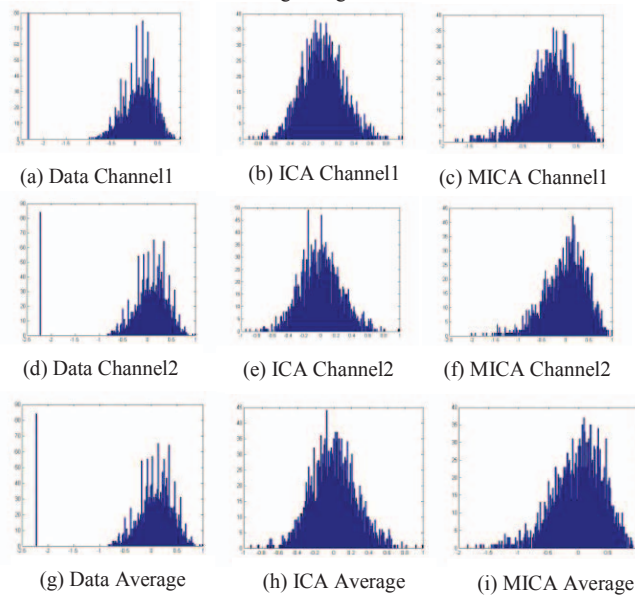


Figure 3. Pigskin Texture: Histograms of the Data channels and the corresponding ICA and MICA distributions. Performance improvement due to MICA = $(KLD(MICA) - KLD(ICA)) / KLD(ICA) = 87.33\%$

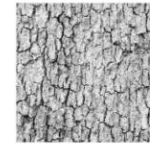


Fig 4: Herringbone

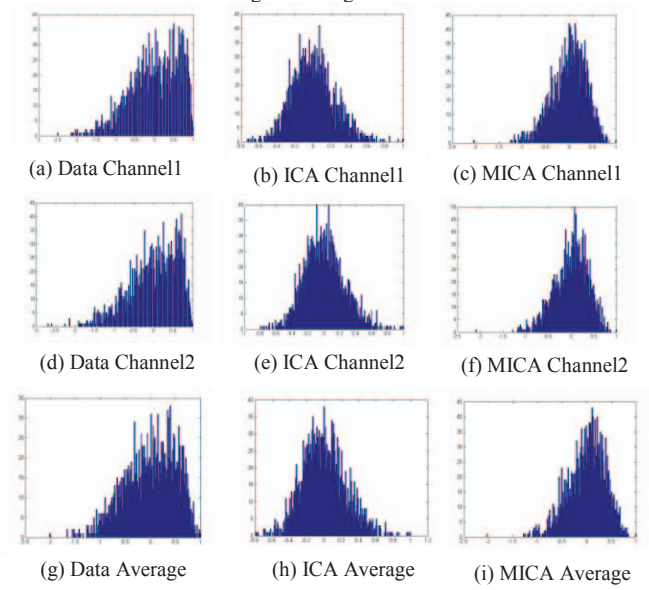


Figure 4. Herringbone Texture: Histograms of the Data channels and the corresponding ICA and MICA distributions. Performance improvement due to MICA = $(KLD(MICA) - KLD(ICA)) / KLD(ICA) = 73.59\%$

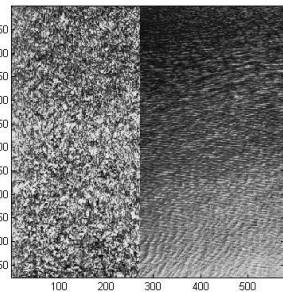


Fig.5(a): Grass-Water multi-texture

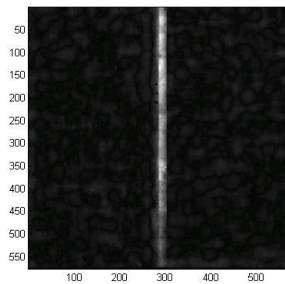


Fig.5(b): NANS map

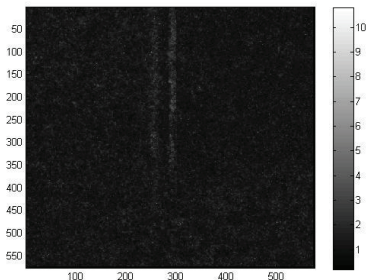


Fig.5(c): MI map