Image Compression Based Upon Independent Component Analysis: Generation of Self-Aligned ICA Bases

Yasuo Matsuyama, Ryo Kawamura, Hiroaki Kataoka, Naoto Katsumata,
Kenzo Tojima, Hideaki Ishijima, and Keita Shimoda

Department of Computer Science, Waseda University
Tokyo, 169-8555 Japan

{yasuo2, ryo@asagi, pp_kataoka@moegi, katsu@ruri,
k-tojima@toki, suzaku@toki, kei215@asagi}.waseda.jp

Abstract

Generation of the ordered set of ICA bases (Independent Component Analysis bases) and its applications to image compression are discussed. The ICA bases have similar properties to existing orthogonal bases. Orthogonal bases generate uncorrelated coefficients, while, the ICA bases bring about independent coefficients. The independence is stronger than the uncorrelatedness. Therefore, the ICA bases can extract source information better. One difficulty using ICA is the permutation indeterminacy among these bases. This paper presents partially supervised learning for generating self-aligned ICA bases. It is observed that: (i) Each basis reflects edges and textures like the early vision. (ii) Bases can be self-aligned in the sense of spatial frequency. (iii) Coefficients of the bases can be used for image compression. Experiments show that (iv) the set of ICA image bases is a well-qualified alternative to existing orthogonal ones.

1. Introduction

Independent Component Analysis (ICA) [1] is a method of multivariate analysis to decompose measured data into independent components. It is a class of learning algorithms from data. Its application is wide including images, speech, music signals and so on. Therefore, ICA has received much attention from communities of adaptive learning and multimedia processing. This paper contributes to these fields by showing a new method to obtain ICA image bases and novel applications to image compression.

Organization of this paper is as follows. In Section 2, the ICA problem is formulated. The role of the ICA basis set is elucidated. The permutation indeterminacy, which essentially exists in the ordinary ICA, is explained. Presentation of the ICA model for image construction is also given. Then, in Section 3, preprocessing, orthonormalization, and ordinary ICA algorithms are explained to assist later explanations on improved methods. In Section 4, The ICA learning with weak guidance is presented. This partial supervision is effective to the reduction of the permutation indeterminacy, whose step is necessary for the application of ICA bases. In Section 5, experiments on digital images are executed. The ICA image bases are successfully aligned by reflecting spatial frequencies. Experiments show that the ICA bases are promising in the image compression as the theory predicts. Section 6 gives concluding remarks with prospects of future studies.

2. Problem Formulation of ICA

2.1. Mixture of Independent Components

In the problem of ICA, a vector random variable

$$\boldsymbol{x} = [x_1, \cdots, x_n]^T \tag{1}$$

is assumed to be generated by another random variable

$$\boldsymbol{s} = [s_1, \cdots, s_n]^T \tag{2}$$

by the following mixture.

$$\boldsymbol{x} = \boldsymbol{A}\boldsymbol{s} = [\boldsymbol{a}_1 \cdots, \boldsymbol{a}_n]\boldsymbol{s} = \textstyle\sum_{i=1}^{n} \boldsymbol{a}_i s_i \tag{3}$$

The matrix $\boldsymbol{A}$ and the vector $\boldsymbol{s}$ are both unknown except for the following information.

(a) The components s_i and s_j are independent each other for $i \neq j$.

(b) The components s_i, $(i = 1, \cdots, n)$, are non-Gaussian except for at most one i.

Under the above conditions, we want to estimate a demixing matrix

$$\boldsymbol{W} = \boldsymbol{\Lambda}\boldsymbol{\Pi}\boldsymbol{A}^{-1} \tag{4}$$

so that the components y_i, $(i = 1, \cdots, n)$, of

$$\boldsymbol{W}\boldsymbol{x} \overset{\text{def}}{=} \boldsymbol{y} = [y_1, \cdots, y_n]^T \tag{5}$$

are independent each other. Here, $\boldsymbol{\Lambda}$ is a nonsingular diagonal matrix which decides components' scale, and $\boldsymbol{\Pi}$ is a permutation matrix. These matrices are unknown too. This property is called the indeterminacy, which essentially exists in the ICA formulation. In this paper, such indeterminacy will be carefully avoided.

2.2. ICA bases

Column vectors of $\boldsymbol{W}^{-1} \overset{\text{def}}{=} \boldsymbol{U}$ can be interpreted as ICA bases since the following equality holds for the observed data $\boldsymbol{x}$.

$$\boldsymbol{x} = \boldsymbol{U}\boldsymbol{y} = [\boldsymbol{u}_1, \cdots, \boldsymbol{u}_n]\boldsymbol{y} = \textstyle\sum_{i=1}^{n} \boldsymbol{u}_i y_i \tag{6}$$

In order to save notational alphabets, $\boldsymbol{U}$ is re-expressed by $\boldsymbol{A}$ hereafter, and so is $\boldsymbol{y}$ by $\boldsymbol{s}$. This is applied only if there is no confusion.

When an ICA basis $\boldsymbol{a}_i$ is used in image processing, it is interpreted as a two dimensional patch $\{\{a_i(x,y)\}_{x=1}^{m}\}_{y=1}^{m}$. Then, each pixel is modeled by

$$I(x,y) = \textstyle\sum_{i=1}^{n} a_i(x,y) s_i, \tag{7}$$

where $n = m^2$. Once the ICA bases are learned from data, they are *fixed*. Therefore, $\{s_i\}_{i=1}^{n}$ are subject to coding for image compression.

3. ICA Learning Algorithms

3.1. Preprocessing and Orthonormalization

Observed data are preprocessed in the following way so that the estimate of $\boldsymbol{W}$ converges properly.

1. [Mean and variance normalization] Observed data are normalized to have the zero mean and the unit variance.

2. [Whitening] Observed data are then transformed to $\boldsymbol{z} = \boldsymbol{V}\boldsymbol{x}$ so that $\mathcal{E}[\boldsymbol{z}\boldsymbol{z}^T] = \boldsymbol{I}$. Here, $\mathcal{E}$ stands for the expectation. We use $\boldsymbol{V} = \boldsymbol{D}^{-1/2}\boldsymbol{E}^T$ in our experiments. Here, $\boldsymbol{D}$ is a diagonal matrix whose elements are eigenvalues of $\mathcal{E}[\boldsymbol{x}\boldsymbol{x}^T]$. $\boldsymbol{E}$ is the matrix whose columns are corresponding eigenvectors.

3. [Orthonormalization] Another transformation is the orthonormalization: $\boldsymbol{W} \leftarrow (\boldsymbol{W}\boldsymbol{W}^T)^{-1/2}\boldsymbol{W}$. This is an expensive computation, however, the merits of $\boldsymbol{U} = \boldsymbol{W}^T$ and $\boldsymbol{W}^T\boldsymbol{W} = \boldsymbol{I}$ are obtained.

3.2. First-Stage Learning Algorithm

Estimation or learning of $\boldsymbol{W}$ from observed data is performed by the following iteration:

$$\boldsymbol{W}^{\text{new}} = f(\boldsymbol{W}^{\text{old}}), \tag{8}$$

or equivalently,

$$\boldsymbol{W}^{\text{new}} = \boldsymbol{W}^{\text{old}} + \varDelta\boldsymbol{W}. \tag{9}$$

The updated vector $\boldsymbol{W}^{\text{new}}$ can be obtained by optimizing statistical measures for the independence [2] $\sim$ [7].

We gave necessary explanations on the first-stage algorithm for $\boldsymbol{W}$ except for the following. We are given sample image patches rather than an abstract random variable in a probability space. Therefore, we need to write down these samples in matrix forms: $\boldsymbol{X} = [\boldsymbol{x}(1), \cdots, \boldsymbol{x}(m)]$, $\boldsymbol{S} = [\boldsymbol{s}(1), \cdots, \boldsymbol{s}(m)]$, and $\boldsymbol{Y} = [\boldsymbol{y}(1), \cdots, \boldsymbol{y}(m)]$. Thus, the data generation model is expressed by

$$\boldsymbol{X} = \boldsymbol{A}\boldsymbol{S}. \tag{10}$$

Then, the first-stage learning algorithm becomes as follows.

[First-stage learning algorithm]

[Step 1 (Preprocessing 1)]
Obtain a sample matrix $\boldsymbol{X}$ as a training data set. Normalize each column vector to be zero mean and unit variance.

[Step 2 (Preprocessing 2: Whitening)]
Obtain Whitening Matrix $\boldsymbol{V}$ from $\boldsymbol{X}$, and compute $\boldsymbol{Z} = \boldsymbol{V}\boldsymbol{X}$.

[Step 3 (Initialization)]
Choose an orthonormalized initial value for $\boldsymbol{W}$.

[Step 4 (Update 1)]
Update $\boldsymbol{W}$ by (8) or (9).

[Step 5 (Update 2)]
Orthonormalize the matrix $\boldsymbol{W}$.

[Step 6 (Convergence check)]

Check to see if convergence is achieved. Otherwise repeat Steps 4 and 5.

[Step 7 (Resulting matrices)]

Resulting matrices are obtained by

$$\boldsymbol{W}_{\text{stage1}} = \boldsymbol{W}\boldsymbol{V}, \tag{11}$$

and

$$\boldsymbol{A}_{\text{stage1}} = (\boldsymbol{W}\boldsymbol{V})^{-1} = \boldsymbol{V}^{-1}\boldsymbol{W}^{T}. \tag{12}$$

It is necessary to comment here that:

(i) The first-stage algorithm still inherits the permutation indeterminacy. We need further learning algorithms which does not suffer from this indeterminacy.

(ii) In the image compression, the matrix

$$\boldsymbol{Y}_{\text{data}} = \boldsymbol{W}_{\text{stage1}}\boldsymbol{X}_{\text{data}} \tag{13}$$

is encoded to $\hat{\boldsymbol{Y}}_{\text{data}}$. Decoded is then

$$\hat{\boldsymbol{X}}_{\text{data}} = \boldsymbol{A}_{\text{stage1}}\hat{\boldsymbol{Y}}_{\text{data}}. \tag{14}$$

4. Learning Under Weak Guidance

4.1. Indeterminacy Reduction I: Topographic Alignment of ICA Bases

The above $\boldsymbol{A}_{\text{stage1}}$ could be used as a set of image compression bases, if one would dare to check manually the whole matrix pattern, and if high performance is not required. Thus, the bases are more suitable for the image compression if they have ordered by spatial frequencies precisely. Therefore, we consider to use the resulting image bases as an initial set for further learning modification. This is allowed since the image bases need not be computed on-line but to be stored in the encoder-decoder pair. There is one more evidence to support this: All computation in this paper can be carried out by a conventional personal computer, which will be understood in Section 5.

The first step to obtain an aligned image basis set is to modify the matrix $\boldsymbol{W}_{\text{stage1}}$ by using the topographic ICA [8]. In this case, (9) is used with the following computation:

$$\Delta\boldsymbol{w}_i = \eta E[\boldsymbol{z}(\boldsymbol{w}_i^T\boldsymbol{z})r_i], \tag{15}$$

$$r_i = \textstyle\sum_{k=1}^{n} h(i,k)G'(\sum_{j=1}^{n} h(k,j)(\boldsymbol{w}_j^T\boldsymbol{z}^2)). \tag{16}$$

On the choices of $G(y)$ and $h(i,j)$, readers are requested refer to [8]. Hereafter, the update matrix by (15) is denoted by $\Delta\boldsymbol{W}_{\text{tp}}$.

4.2. Indeterminacy Reduction II: Weak Guidance

Resulting ICA bases as a topographic map show an intriguing visual pattern. But, a very important indeterminacy is not yet resolved. A human can instantly find the position of the central basis corresponding to the lowest spatial frequency, however, machines can not do so instantly. Therefore, we need a further important mechanism to reduce such indeterminacy. This is the method of *weak guidance* as a partially supervised learning. Such a method was first used in the distillation of brain maps from fMRI data [9], [10].

[Weak Guidance]

First, we prepare a teacher signal, or a reference pattern, as a matrix $\bar{\boldsymbol{R}}$. Then, we compute $\boldsymbol{U} = \boldsymbol{V}^{-1}\boldsymbol{W}^{T}$. The increment by the teacher signal is

$$\Delta\boldsymbol{U} = \boldsymbol{V}\{\lambda(\bar{\boldsymbol{R}} - \boldsymbol{U})\}. \tag{17}$$

Here, λ is a learning parameter. Then, the update term for the weak guidance is computed by

$$\Delta\boldsymbol{W}_{\text{wg}} = -\boldsymbol{W}\Delta\boldsymbol{U}\boldsymbol{W}. \tag{18}$$

Readers are requested refer to [9] or [10] for the derivation of (17) and (18).

4.3. Total Learning Algorithm

By the preceding preparations, the total algorithm to obtain the ICA bases can be described as follows.

[Step 1 (Learning parameters)]

Control rules of the small learning parameters $\eta > 0$ and $\lambda > 0$ are decided. The rules can be arbitrary as long as (i) η increases and saturates. (ii) λ decreases.

[Step 2 (Weak guidance)]

Compute the updated matrix with the weak guidance

$$\boldsymbol{W} \leftarrow \boldsymbol{W} + \Delta\boldsymbol{W}_{\text{wg}}. \tag{19}$$

[Step 3 (Topographical map)]

Compute the updated matrix with the topographic constraint

$$\boldsymbol{W} \leftarrow \boldsymbol{W} + \Delta\boldsymbol{W}_{\text{tp}}. \tag{20}$$

[Step 4 (Convergence check)]

Check to see if the matrix update is converged. If not, then the iteration is repeated on Steps 2 and 3 after the update of λ, η, and $\boldsymbol{W}$.

5. Experimental Results

5.1. ICA Image Bases with Self-Alignment

All necessary tools were given in the preceding sections. We can now apply them to real images. Training data for the ICA bases contain many images such as natural images, screen text images, and animations. Figure 1 illustrates the ICA bases obtained by the topographic method alone, i.e., without the weak guidance. Each basis is of 8 × 8 pixels so that the size is compatible with usual JPEG and JPEG2000. As can be observed, the basis of the lowest spatial frequency is located off-centered in the two dimensional array. This position can not be specified in advance. Therefore, the human perception is still necessary to identify where the exact center is. Besides, the obtained ICA bases are inefficient since the center is near the corner of the array.

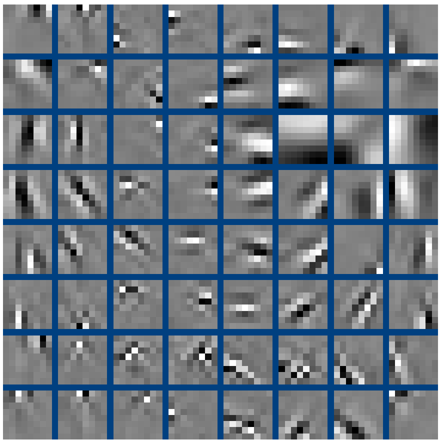


Figure 1. Image bases only by the topographic method.

Figure 2 shows the resulting self-aligned ICA bases by our weak guidance method. The first basis is located at the north west of the four central bases. Low-frequency bases are concentrated around the center of the two-dimensional array. High-frequency bases are located at the corners. This was specified to be so by virtue of the weak guidance. We call such a class of bases the *ICA ripplet set*, or simply the *ripplet set*. The ripplet set is readily applicable to the image compression due to the following properties.

(a) Ordering from low to high spatial frequencies is completed.

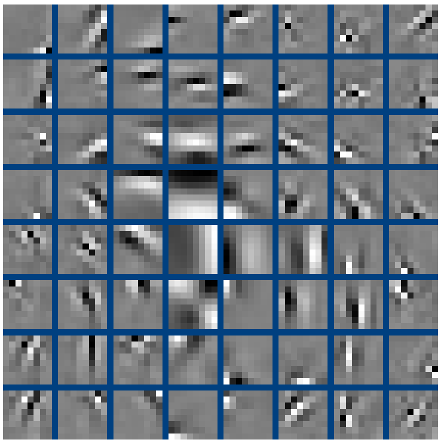


Figure 2. Self-aligned image bases by the weak guidance.

(b) Bases are balanced because of the centering by the weak guidance.

Item (a) can be observed from Figure 3 which was obtained from Figure 2 by the clockwise spiral sorting started from the origin, the north west of the four central bases. This figure clearly shows a self-aligned ordering from low to high spatial frequencies which has the following merit:

(c) Users of this set can understand the role of each basis in a linearly ordered sense. High-frequency bases may correspond to noisy patterns. Such a merit will be utilized in Section 5.3.

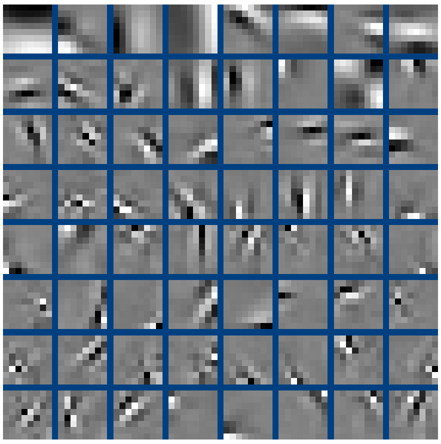


Figure 3. Aligned image bases.

Figures 1 to 3 can claim the similarity to the receptive field properties [11]. Not only appreciating such an intriguing similarity, but we pay attention to the self-aligned ICA bases of Figures 2 and 3 on the use for the image compression.

5.2. Distribution of Coefficients

Source images are reconstructed by using Equation (7). We remind readers here that the relationship on the ICA image bases: $\boldsymbol{A}^{-1} \leftarrow \boldsymbol{W}$. Since the image bases are not altered any more, they are memorized in the encoder and the decoder. In the encoder, coefficients s_i are obtained by $\boldsymbol{S} = \boldsymbol{A}^{-1}\boldsymbol{X}$. Here, each column vector of $\boldsymbol{X}$ corresponds to an image patch. Then, encoded values of elements s_i in $\boldsymbol{S}$ are transmitted (in a two dimensional form, $s_{i,j}$).

The distribution of $\boldsymbol{s}$ is $\{8 \times 8 = 64\}$-dimensional which is unable to illustrate visually. But, the flatness of the distribution can be estimated from the histogram of $s_{i,j}$. If the distribution of $s_{i,j}$ were nearly flat, there would be very little possibility for data compression because of high entropy. Therefore, we have to examine the distribution on real images. Figure 4 is the resulting distribution obtained from an image outside of the training data. The horizontal axis shows values of s_i. The vertical axis shows the number of appearances, i.e., the frequency. As can be observed, this distribution is far from flat. It is highly super-Gaussian reflecting the nature of the ICA transformation of images. This means that most of s_i are centered around zero. Few important numbers are distant from zero. Therefore, we can judge that the distribution of $\boldsymbol{s}$ is very sparse. This property encourages us with the anticipation that the encoding for data compression will be effective.

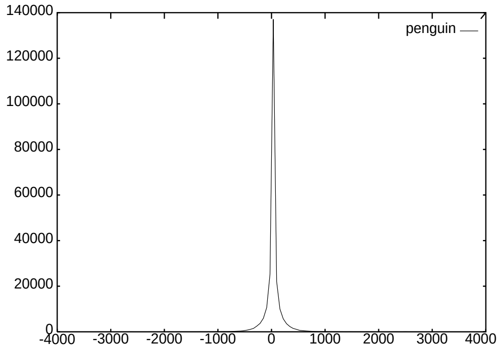


Figure 4. Distribution of coefficients to be encoded.

5.3. Image Compression

Here, we discuss the case of variable-length coding based upon the run-length and Huffman coding. The source to be compressed is the matrix

$$\boldsymbol{S} = [\boldsymbol{s}_1, \cdots, \boldsymbol{s}_M]. \tag{21}$$

Here, $\boldsymbol{s}_i$ is the vector coefficient for one patch in the source image. Therefore, $M = 3750$ for a 600×400 pixel image since $(600/8) \times (400/8) = 3750$. The vector $\boldsymbol{s}_i$ is quantized in group. The quantization is set to be granular if a coefficient is for a low spatial frequency. On the other hand, the quantization is rough if the coefficient is for higher frequencies. Quantized zeroes appear frequently because of the sparseness explained in Section 5.2. Then, we denote the resulting coefficient matrix by $\tilde{\boldsymbol{S}}$. We found that quantized zeroes run consecutively if we raster scan this $\tilde{\boldsymbol{S}}$ *vertically* because of the property explained in Item (c) of Section 5.1: High-frequency bases correspond to noisy patterns. Therefore, run-length coding is effective. Huffman coding is used for non-zeroes.

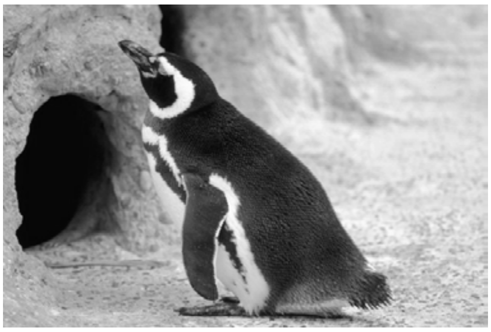


Figure 5. Uncompressed image.

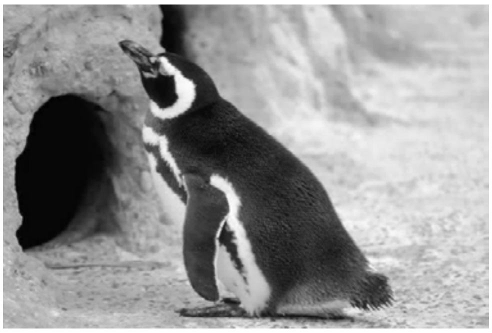
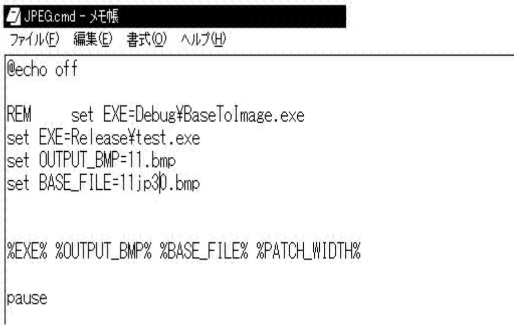


Figure 6. Compressed image 1.

Figures 5 is a source image selected from [12]. Figure 6 is a compressed image by this paper's method. The compressed image has the performance of $\mathrm{SNR}_{\mathrm{pp}} =$



```
@echo off

REM     set EXE=Debug¥BaseToImage.exe
set EXE=Release¥test.exe
set OUTPUT_BMP=11.bmp
set BASE_FILE=11jp30.bmp


%EXE% %OUTPUT_BMP% %BASE_FILE% %PATCH_WIDTH%

pause
```

Figure 7. Compressed image 2.

35.2 dB at 1.24 bit/pixel. Figure 7 is another compressed image containing charecters. This image is not a set of outline fonts but is obtained from a computer display. This image has the performance of $\mathrm{SNR}_{\mathrm{pp}} = 34.8$ dB at 1.28 bit/pixel.

More experiments besides Figures 6 and 7 were tried. We can conclude that the image compression based upon the ICA bases designed by this paper's method shows the excellent performance.

6. Concluding Remarks

The main purpose of this paper was to show that

(i) The permutation indeterminacy of the ICA can be avoided. The resulting bases can be used in engineering applications, particularlly for image compression.

(ii) The ICA bases learned from images extract important information. Such bases can be applied to reconstruct unlearned images.

(iii) Coefficients for the reconstruction can be used for the image compression.

It was possible to show that the image compression based upon the ICA bases is promising. We shall have the immediate sophistication of this paper's study as follows:

(a) Incorporation of better lossless coding on coefficients.

(b) Applications to color image compression.

Acknowledgment

This study was supported by the Grant-in-Aid for Scientific Research #15300077 and by the Productive ICT Academia of the 21st Century COE Program granted to Waseda University. The authors are grateful to Mr. S. Imahara of Toshiba Co. for his early contributions.

References

[1] C. Jutten and J. Herault, "Blind separation of sources, part I: An adaptive algorithm based on neuromimetic architecture," Signal Processing, vol. 24, pp. 1-10, 1991.

[2] J-F. Cardoso and A. Souloumiac, "Blind beamforming for non Gaussian signals," IEE Proceedings F, vol. 140, pp. 362-370, 1993.

[3] A. J. Bell and T. J. Sejnowski, "An information-maximization approach to blind separation and blind deconvolution," Neural Computation, vol. 7, pp. 1129-1159, 1995.

[4] H. H. Yang and S. Amari, "Adaptive online learning algorithm for blind separation: Maximum entropy and minimum mutual information," Neural Computation, vol. 9, pp. 1457-1482, 1997.

[5] A. Hyvärinen "Fast and robust fixed-point algorithms for independent component analysis," IEEE Trans. Neural Networks, vol. 10, pp. 626-639, 1999.

[6] Y. Matsuyama, N. Katsumata, Y. Suzuki and S. Imahara, "The α-ICA algorithm," Proc. Int. Workshop on Independent Component Analysis, pp. 297-302, 2000.

[7] Y. Matsuyama, S. Imahara and N. Katsumata, "Optimization transfer for computational learning," Proc. Int. Joint Conf. on Neural Networks, vol. 3, pp. 1883-1888, 2002.

[8] A. Hyvärinen, P.O. Hoyer and M. Inki, "Topographic independent component analysis," Neural Computation, vol. 13, pp. 1527-1558, 2001.

[9] Y. Matsuyama and S. Imahara, "The α-ICA algorithm and brain map distillation from fMRI images," Proc. Int. Conf. Neural Info. Processing, vol. 2, pp. 708-713, 2000.

[10] Y. Matsuyama, N. Katsumata anr R. Kawamura, "Independent component analysis minimizing convex divergence," Proc. ICANN/ICONIP03, Lecture Notes in Computer Science, Springer-Verlag, 2003.

[11] B. A. Olshausen and D. J. Field, "Emergence of simple-cell receptive field properties by learning a sparse code for natural images," Nature, vol. 381, pp. 607-609, 1996.

[12] "Master Photo 75,000," H^2 Soft Co., 1999.