

Unsupervised Image Classification, Segmentation, and Enhancement Using ICA Mixture Models

Te-Won Lee and Michael S. Lewicki

Abstract—An unsupervised classification algorithm is derived by modeling observed data as a mixture of several mutually exclusive classes that are each described by linear combinations of independent, non-Gaussian densities. The algorithm estimates the data density in each class by using parametric nonlinear functions that fit to the non-Gaussian structure of the data. This improves classification accuracy compared with standard Gaussian mixture models. When applied to images, the algorithm can learn efficient codes (basis functions) for images that capture the statistically significant structure intrinsic in the images. We apply this technique to the problem of unsupervised classification, segmentation, and denoising of images. We demonstrate that this method was effective in classifying complex image textures such as natural scenes and text. It was also useful for denoising and filling in missing pixels in images with complex structures. The advantage of this model is that image codes can be learned with increasing numbers of classes thus providing greater flexibility in modeling structure and in finding more image features than in either Gaussian mixture models or standard independent component analysis (ICA) algorithms.

Index Terms—Blind source separation, denoising, fill-in missing data, Gaussian mixture model, image coding, independent component analysis, maximum likelihood, segmentation, unsupervised classification.

I. INTRODUCTION

MODELING the statistical relations in images is an important framework for image processing and synthesis algorithms [7], [10], [39]. In many applications a fixed representation such as the Fourier transformation is assumed to model a large number of different images. Image processing techniques that use a more flexible model that is adapted to the structure of the underlying data can achieve better results. Adaptive techniques such as the principal component analysis (PCA) approximate the intrinsic structure of image data up to its second order statistics. The PCA representation has been known to result in a compact representation when applied to images of natural scenes [18]. Recently, several methods have been proposed to learn image codes that utilize a set of linear basis functions. Olshausen and Field [36] used a sparseness criterion and found codes that were similar to localized and ori-

ented receptive fields found in V1. Similar results were obtained by Bell and Sejnowski [5] and Lewicki and Olshausen [32] using the infomax ICA algorithm and a Bayesian approach respectively. These results support Barlow's proposal [3] that the goal of sensory is to transform the input signals such that it reduces the redundancy between the inputs. These recent approaches have in common that they try to reduce the information redundancy by capturing the statistical structure in images that is beyond second order information. Independent component analysis (ICA) is a technique that exploits higher-order statistical structure in data. This method has recently gained attention due to its applications to signal processing problems including speech enhancement, telecommunications and medical signal processing. ICA finds a linear nonorthogonal coordinate system in multivariate data determined by second- and higher-order statistics. The goal of ICA is to linearly transform the data such that the transformed variables are as statistically independent from each other as possible [4], [11], [14], [23], [25]. ICA generalizes PCA and, like PCA, has proven a useful tool for finding structure in data.

In this paper, we are interested in finding statistically significant structures in images. Images may be constructed by classes of image types, such as text overlapping with natural scenes or the natural scene itself may have diverse structures or textures such as trees and rocks. We model the underlying image with a mixture model that can capture the different types of image textures with classes. Each class is learned in an unsupervised fashion and contains the statistical intrinsic structure of its image texture. In a mixture model (see for example, [16]), the observed data can be categorized into several mutually exclusive classes. When the data in each class are modeled as multivariate Gaussian, it is called a Gaussian mixture model. We generalize this by assuming that the data in each class are generated by a linear combination of independent, non-Gaussian sources, as in the case of ICA. We call this model an ICA mixture model. This allows modeling of classes with non-Gaussian structure, e.g., platykurtic or leptokurtic probability density functions. The algorithm for learning the parameters of the model uses gradient ascent to maximize the log likelihood function. We apply this learning algorithm to the problem of unsupervised classification, segmentation and denoising of images.

This paper is organized as follows. We present the ICA mixture model and show how to infer the parameters for this model. Detailed derivations of the learning algorithm are in [24], [27], and [29]. Section III shows how this method can be used to learn codes for images of different types. Section IV uses the learned codes to classify and segment individual images with complex structure. Section V extends this model to denoising images and

Manuscript received May 10, 2000; revised November 14, 2001. T.-W. Lee was supported by the Swartz Foundation. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Bangalore S. Manjunath.

T.-W. Lee is with the Institute for Neural Computation, University of California at San Diego, La Jolla, CA 92093 USA (e-mail: tewon@salk.edu).

M. S. Lewicki is with the Department of Computer Science and Center for the Neural Basis of Cognition, Carnegie Mellon University, Pittsburgh, PA 15213 USA (e-mail: lewicki@cnbc.cmu.edu).

Publisher Item Identifier S 1057-7149(02)01735-9.

filling in missing pixels in images. Finally, Section VI relates these methods to other algorithms and gives future directions of this line of research.

II. ICA MIXTURE MODEL

Assume that the data $\mathbf{X} = \{\mathbf{x}_1, \dots, \mathbf{x}_t, \dots, \mathbf{x}_T\}$ are drawn independently and generated by a mixture density model [16]. T is the total number of data vectors and each data vector \mathbf{x}_t is an N -dimensional data vector where N is assumed to be the number of sensors. The likelihood of the data is given by the joint density

$$p(\mathbf{X} | \Theta) = \prod_{t=1}^T p(\mathbf{x}_t | \Theta). \quad (1)$$

The mixture density is

$$p(\mathbf{x}_t | \Theta) = \sum_{k=1}^K p(\mathbf{x}_t | C_k, \theta_k) p(C_k) \quad (2)$$

where $\Theta = (\theta_1, \dots, \theta_K)$ are the unknown parameters for each $p(\mathbf{x} | C_k, \theta_k)$, called the component densities. C_k denotes the class k and it is assumed that the number of classes, K , are known in advance. However, we can estimate the number of classes with a Bayesian method using a split and merge algorithm as described in [2]. Assume that the component densities are non-Gaussian and the data within each class are described by

$$\mathbf{x}_t = \mathbf{A}_k \mathbf{s}_k + \mathbf{b}_k \quad (3)$$

where \mathbf{A}_k is a $N \times M$ scalar matrix¹ and \mathbf{b}_k is the bias vector for class k . The vector \mathbf{s}_k is called the source vector² (these are also the coefficients for each basis function). Note that (3) shows one way of K ways for generating the data vector \mathbf{x}_t . Depending on the values for $\mathbf{A}_k, \mathbf{s}_k, \mathbf{b}_k$ there are K ways for viewing \mathbf{x}_t . However, we assume mutually exclusive classes and maximum likelihood estimation results in one model that fits the data the best.

It is assumed that the individual sources $s_{k,i}$ within each class are mutually independent across a data ensemble. For simplicity, we consider the case where the number of sources (M) is equal to the number of linear combinations (N). The motivation for this equality is due to a simpler calculation of the learning rules since an exact inverse exist for \mathbf{A}_k . In case of $N < M$ an over-complete representation is required [33] which is computationally quite burdensome and not necessary for our purpose. For $N > M$, there are ICA subspace methods [12], [21] which require knowledge about the optimal subspace dimensionality which is very application and data dependent.

Fig. 1 shows a simple example of a dataset describable by an ICA mixture model. Each class was generated from (3) using a different \mathbf{A}_k and \mathbf{b}_k . Class “o” was generated by two uniformly distributed sources, whereas class “+” was generated by two Laplacian distributed sources ($p(s) \propto \exp(-|s|)$). The task is to classify the unlabeled data points and to determine the pa-

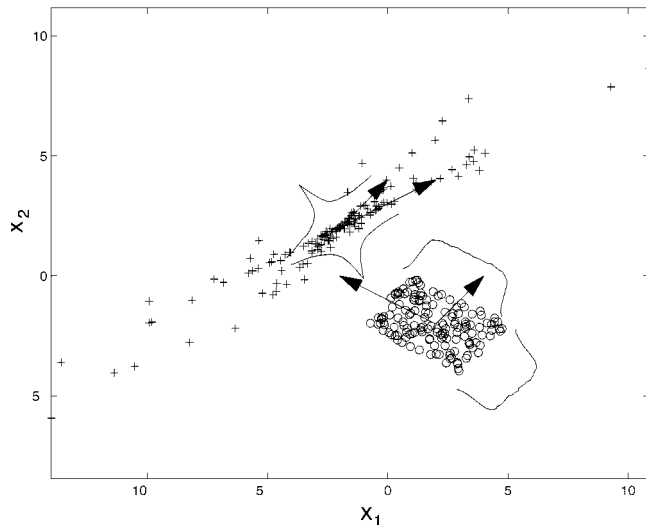


Fig. 1. Simple example for classifying an ICA mixture model. There are two classes, “+” and “o”; each class was generated by two independent variables, two bias terms and two basis vectors. Class “o” was generated by two uniformly distributed sources as indicated next to the data class. Class “+” was generated by two Laplacian distributed sources with a sharp peak at the bias and heavy tails. The inset graphs show the distributions of the source variables, $s_{i,k}$, for each basis vector.

rameters for each class, $(\mathbf{A}_k, \mathbf{b}_k)$ and the probability of each class $p(C_k | \mathbf{x}_t, \Theta)$ for each data point.

The iterative learning algorithm which performs gradient ascent on the total likelihood of the data in (2) has the following steps.

- Compute the log-likelihood of the data for each class

$$\log p(\mathbf{x}_t | C_k, \theta_k) = \log p(\mathbf{s}_k) - \log(\det |\mathbf{A}_k|) \quad (4)$$

where $\theta_k = \{\mathbf{A}_k, \mathbf{b}_k\}$. Note that \mathbf{s}_k is implicitly modeled for the adaptation of \mathbf{A}_k .

- Compute the probability for each class given the data vector \mathbf{x}_t

$$p(C_k | \mathbf{x}_t, \Theta) = \frac{p(\mathbf{x}_t | \theta_k, C_k) p(C_k)}{\sum_k p(\mathbf{x}_t | \theta_k, C_k) p(C_k)}. \quad (5)$$

- Adapt the basis functions \mathbf{A}_k and the bias terms \mathbf{b}_k for each class. The basis functions are adapted using gradient ascent

$$\begin{aligned} \Delta \mathbf{A}_k &\propto \frac{\partial}{\partial \mathbf{A}_k} \log p(\mathbf{x}_t | \Theta) \\ &= p(C_k | \mathbf{x}_t, \Theta) \frac{\partial}{\partial \mathbf{A}_k} \log p(\mathbf{x}_t | C_k, \theta_k). \end{aligned} \quad (6)$$

This gradient can be approximated using an ICA algorithm, as shown below. The gradient can also be summed over multiple data points. An approximate update rule was used for the bias terms (see [27], [29], for an on-line update version for \mathbf{b}_k and the derivations)

$$\mathbf{b}_k = \frac{\sum_t \mathbf{x}_t p(C_k | \mathbf{x}_t, \Theta)}{\sum_t p(C_k | \mathbf{x}_t, \Theta)} \quad (7)$$

where t is the data index ($t = 1, \dots, T$).

¹This matrix is called the mixing matrix in ICA papers and specifies the linear combination of independent sources. Here, we refer to \mathbf{A} as the basis matrix to distinguish this from the word mixture in the mixture model.

²Note that we have omitted the data index t for $\mathbf{s}_{k,t}$.

The gradient of the log of the component density in (6) can be approximated using an ICA model. There are several methods for adapting the basis functions in the ICA model [4], [11], [14], [21], [26]. One of the differences between the ICA algorithms are the use of higher-order statistics such as cumulants versus models that use a predefined density model. In our model, we are interested in iteratively adapting the class parameters and modeling a wider range of distributions. The extended infomax ICA learning rule which is able to blindly separate unknown sources with sub- and super-Gaussian distributions.³ This is achieved by using a simple type of learning rule first derived by [20]. The learning rule in [26] uses the stability analysis of [11] to switch between sub- and super-Gaussian regimes

$$\Delta \mathbf{A}_k \propto -p(C_k | \mathbf{x}_t, \Theta) \mathbf{A}_k [\mathbf{I} - \mathbf{K} \tanh(\mathbf{s}_k) \mathbf{s}_k^T - \mathbf{s}_k \mathbf{s}_k^T] \quad (8)$$

where \mathbf{K} is an $N \times N$ dimensional diagonal matrix and k_i are elements in the diagonal that indicate if the source s_i is sub-Gaussian or super-Gaussian. The k_i 's can be derived from the generic stability analysis [26]. $\mathbf{W}_k = \mathbf{A}_k^{-1}$ is called the filter matrix. The adaptation of the source density parameters are the $k_{k,i}$'s [26]

$$k_{k,i} = \text{sign} \left(E\{\text{sech}^2(s_{k,i})\} E\{s_{k,i}^2\} - E\{[\tanh(s_{k,i})s_{k,i}]\} \right). \quad (9)$$

The source distribution is super-Gaussian when $k_{k,i} = 1$ and sub-Gaussian when $k_{k,i} = -1$. For the log-likelihood estimation in (4) the term $\log p(\mathbf{s}_k)$ can be approximated as follows:⁴

$$\log p(\mathbf{s}_k) \propto - \sum_{i=1}^N \left(k_{k,i} \log(\cosh s_{k,i}) - \frac{s_{k,i}^2}{2} \right). \quad (10)$$

Super-Gaussian densities, are approximated by a density model with heavier tail than the Gaussian density; sub-Gaussian densities are approximated by a bimodal density [20]. This source density approximation is adequate for most problems [26].⁵ The extended infomax algorithm is used for finding the parameters in Fig. 1. A continuous parameter is inferred that fits a wide range of distributions.

When only sparse representations are needed, a Laplacian prior [$p(s) \propto \exp(-|s|)$] can be used for the weight update, which simplifies the infomax learning rule

$$\Delta \mathbf{A}_k \propto -p(C_k | \mathbf{x}_t, \Theta) \mathbf{A}_k [\mathbf{I} - \text{sign}(\mathbf{s}_k) \mathbf{s}_k^T], \quad (11)$$

$$\log p(\mathbf{s}_k) \propto - \sum_i |s_{k,i}| \quad \text{Laplacian prior}$$

³A distribution that is more sharply peaked than a Gaussian around the mean and has heavier tails are called super-Gaussians (leptokurtic distributions) and a distribution with flatter peak such as a uniform distribution is called sub-Gaussian (platykurtic distribution).

⁴The indices k, i, t stand for class index, source index and time index. In case of $s_{k,t}$, the source vector is estimated for class k and in case of $s_{k,i}$ the source coefficient i is estimated for class k .

⁵Recently, we have replaced this with a more general density using an exponential power distribution [27].

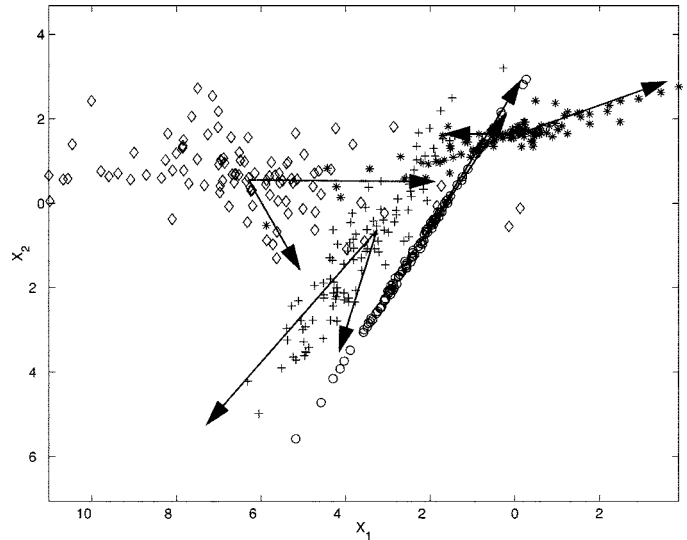


Fig. 2. Example of classification of a mixture of non-Gaussian densities, each describable as a linear combination of independent components. There are four different classes, each generated by two randomly chosen independent variables and bias terms. The algorithm is able to find the independent directions (basis vectors), bias vectors and the source density parameters for each class.

Note that the above adaptation rule is the learning rule that we applied to learning the image features. The use of other ICA algorithms is detailed in [24] and [29]. Although the Laplacian prior was imposed on the coefficients, a more flexible prior such as the generalized Gaussian density model [9], [27], [34], was applied to the same data set. The results were very similar to the results with the Laplacian prior ICA suggesting that enforcing independence among the outputs yields in sparse source densities.

A. Unsupervised Classification Example

To demonstrate the performance of the learning algorithm, we generated random data drawn from different classes and used the proposed method to learn the parameters and to classify the data. Fig. 2 shows an example of four classes in a two-dimensional data space. Each class was generated using random choices for the class parameters. The parameters for the mixture model were inferred using (4)–(7). For this example, the density model parameters for each class were learned using the extended infomax ICA in (8) and (9). For the implementation of the ICA mixture model see Appendix A. The classification was tested by processing each instance with the learned parameters \mathbf{A}_k and \mathbf{b}_k . For this example, in which the classes had several overlapping areas, the classification error on the whole data set averaged over ten trials was $4.8\% \pm 0.7\%$. The Gaussian mixture model used in AutoClass [38] gave an error of $7.8\% \pm 0.5\%$ and converged in all ten trials. For the k -means (Euclidean distance measure) clustering algorithm, the error was 25.3%. The classification error with the original parameters was 3.8%.

III. LEARNING EFFICIENT CODES FOR IMAGES

Recently, several methods have been proposed to learn image codes that utilize a set of linear basis functions. The generative image model assumes a fixed set of basis functions and source coefficients that activate the basis functions to generate a small



Fig. 3. Example of natural scene and text image. The 12×12 pixel image patches were randomly sampled from the images and used as inputs to the ICA mixture model.

patch in the image. Olshausen and Field [36] used a sparseness criterion imposed on the statistical structure of the sources and found codes that were similar to localized and oriented receptive fields. Similar results were presented by Bell and Sejnowski [5] using the infomax ICA algorithm and by Lewicki and Sejnowski [30] using a Bayesian approach. By applying the ICA mixture model we present results that show a higher degree of flexibility in encoding the images. We used images of natural scenes obtained from [36] and text images of scanned newspaper articles. The data set consisted of 12×12 pixel patches ($N = 144$) selected randomly from both image types. Fig. 3 illustrates examples of those image patches. For learning, the means of the data components were subtracted and the components were scaled to unit variance. This implies that there is no need to estimate the bias vectors in case of the image applications presented here. Two complete set of basis functions \mathbf{A}_1 and \mathbf{A}_2 were randomly initialized. Then, for each gradient in (6), a step-size was computed as a function of the amplitude of the basis vectors and the number of iterations. The algorithm converged after $T = 100\,000$ iterations and learned two classes of basis functions. Fig. 4 (top) shows the learned basis functions corresponding to natural images. The basis functions show Gabor-like⁶ structure as previously reported [5], [30], [36]. The similarity to Gabor functions is very close and has been measured in [33]. However, the basis functions corresponding to text images [Fig. 4 (bottom)] resemble bars with different lengths and widths that capture the high-frequency structure present in the text images. Note that unlike the case in k -means clustering or clustering with spherical Gaussians, the classes can be spatially overlapping. In the example of the natural images and newspaper text, both classes had zero mean and the pattern vectors were only distinguished by their relative probabilities under the different classes.

⁶A Gaussian modulated sinusoid.

IV. UNSUPERVISED IMAGE CLASSIFICATION AND SEGMENTATION

In the previous section, we applied the ICA mixture model to learn two classes of basis functions for newspaper text images and images of natural scenes. The same approach can be used to identify multiple classes in a single image. The learned classes are mutually exclusive and by dividing the whole image into small image patches and classifying them we can identify a cluster of patches which encode a certain region or texture of the image.

One example illustrates how the algorithm can learn textures in images by unsupervised classification and therefore is able to segment the image into different classes. The example shows the segmentation of a scanned page from a scientific magazine. This page contains text with an image. In contrast to supervised techniques [8] our method works unsupervised and may be more flexible to a wider range of image types. Fig. 5(a) shows the scanned page with a natural scene and text overlapping with parts of the tree. Fig. 5(b) shows the classification result of the ICA mixture model for two classes using image patches of size 8×8 pixels patches. The model classifies foreground as one class and background as the other class. Although one might expect that classifying text and the natural scene separately would be more efficient the model has no prior information and classifies the image into background and foreground. Fig. 5(c) shows the classification result of 8×8 pixel image patches this time each patch was shifted by one pixel. The resolution is much higher than in Fig. 5(b) and the three classes are background, natural scene and text. Due to the high resolution, background within the text class are automatically segmented. Note that overlapping areas of text and tree are classified as well. The learned basis functions for the three classes reflect the different statistical structures for each

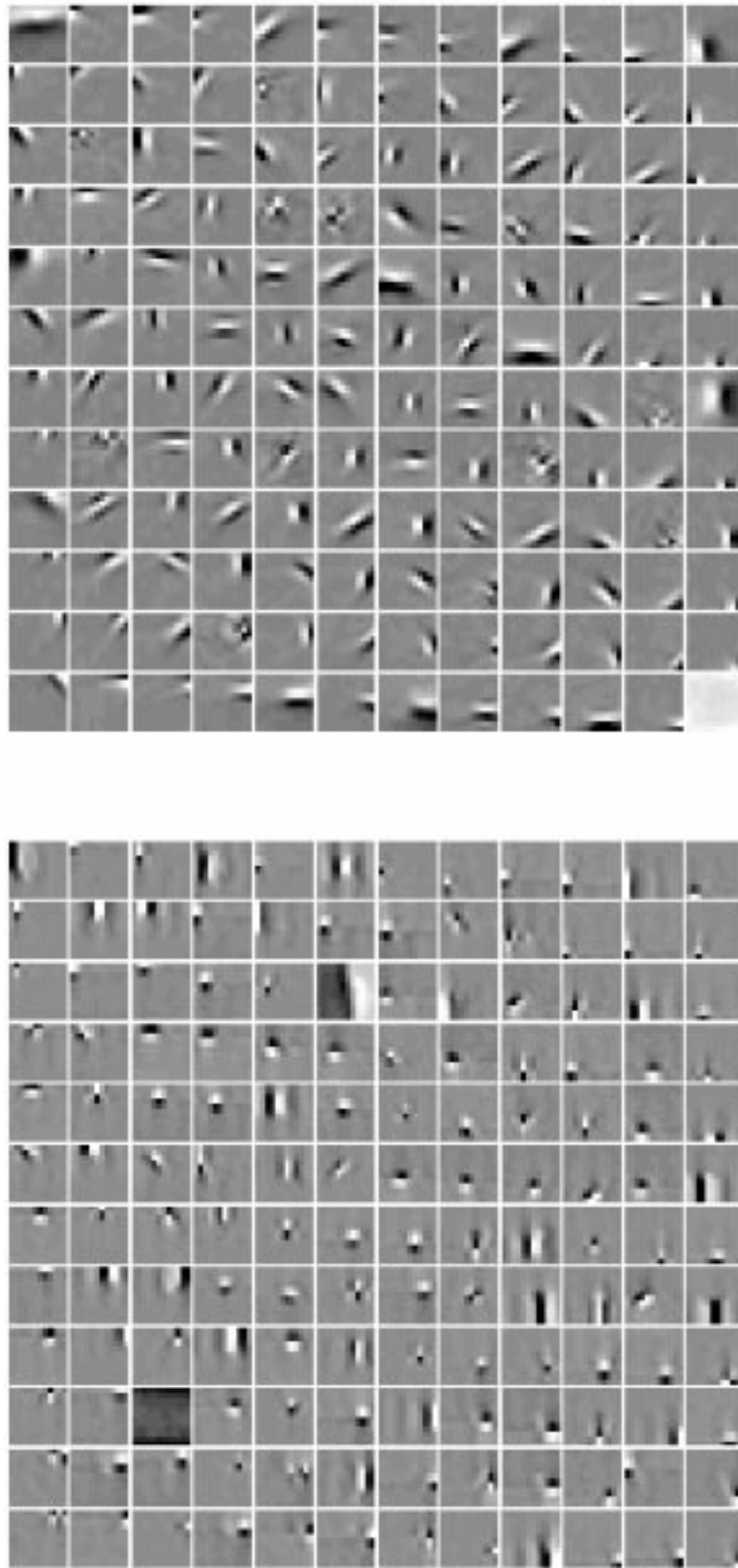


Fig. 4. (Top) Basis function class corresponding to natural images. (Bottom) Basis function class corresponding to text images.

class. In some cases it misclassified the very dark region of the tree as background since this region does not contain enough texture information. To circumvent this problem in real appli-

cations, the model needs to average out the small individual misclassified patches by taking a majority vote over the region or averaging it over the classes.

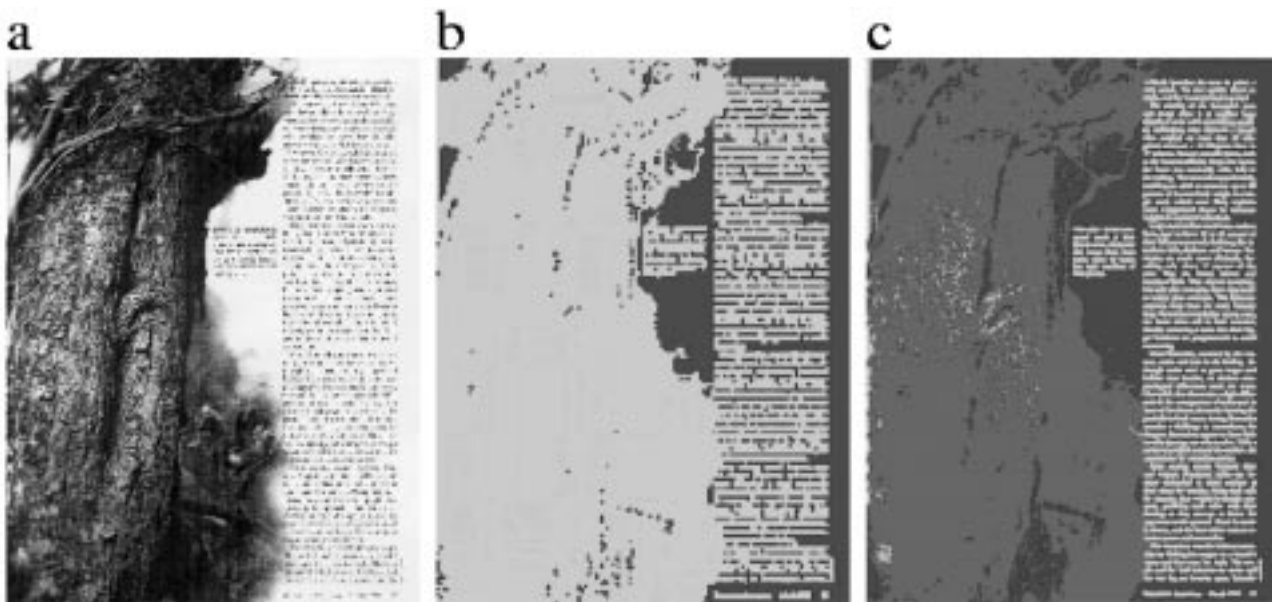


Fig. 5. (a) Example of a natural scene with text. (b) The classification of image patches (8×8 pixels) shifted by eight pixels using the learned two sets of basis functions. (c) The classification of image patches (8×8 pixels) shifted by one pixel using the learned three sets of basis functions.

V. IMAGE ENHANCEMENT

The ICA mixture model provides a good framework for encoding different image types. The learned basis functions can be used for denoising images and filling in missing pixels. Each image patch is assumed to be a linear combination of basis functions plus additive noise: $\mathbf{x}_t = \mathbf{A}_k \mathbf{s}_k + \mathbf{n}$. Our goal is to infer the class probability of the image patch as well as the coefficients \mathbf{s}_k for each class that generate the image. Thus, \mathbf{s}_k is inferred from \mathbf{x}_t by maximizing the conditional probability density $p(\mathbf{s}_k | \mathbf{A}_k, \mathbf{x}_t)$ as shown for a single class in Lewicki and Olshausen [32]

$$\hat{\mathbf{s}}_k = \max_{\mathbf{s}_k} [\log p(\mathbf{x}_t | \mathbf{A}_k, \mathbf{s}_k) + \log p(\mathbf{s}_k)] \quad (12)$$

$$= \min_{\mathbf{s}_k} \left[\frac{\lambda_k}{2} |\mathbf{x}_t - \mathbf{A}_k \mathbf{s}_k|^2 + \alpha_k^T |\mathbf{s}_k| \right] \quad (13)$$

where α_k is the width of the Laplacian p.d.f. and $\lambda_k = 1/\sigma_{k,n}^2$ is the precision of the noise for each class. The inference model in (13) computes the coefficients $\hat{\mathbf{s}}_k$ for each class \mathbf{A}_k , reconstructs the image using $\hat{\mathbf{x}}_t = \mathbf{A}_k \hat{\mathbf{s}}_k$, and computes the class probability $p(C_k | \mathbf{A}_k, \hat{\mathbf{x}}_t)$. For signal-to-noise ratios above 20 dB the mis-classification of image patches was less than 2%. However, the error rate was higher when the noise variance was half the variance of the signal.

To demonstrate how well the basis functions capture the structure of the data we applied the algorithm to the problem of removing noise in two different image types. In Fig. 6(a), a small image was taken from a natural scene and a newspaper text. The whole image was corrupted with additive Gaussian noise that had half of the variance of the original image. The Gaussian noise changes the statistics of the observed image such that the underlying coefficients \mathbf{s} are less sparse than the original data. By adapting the noise level it is possible to infer the original source density by using (13). The adaptation using

the ICA mixture model is better suited for this type of problem than the standard ICA model because the ICA mixture model allows to switch between different image models and therefore is more flexible in reconstructing the image. In this example, we used the two basis functions learned from natural scenes and newspaper text. For denoising, the image was divided into small 12×12 pixel image patch. Each patch was first denoised within each class and then classified by comparing the likelihood of the two classes. In some image processing applications pixel values may be missing. This problem is similar to the denoising problem and the ICA mixture model can be used as a technique to solve this problem. In filling in missing pixels, the missing information can be viewed as another form of noise. Fig. 6 shows an example for denoising and filling in missing pixels using our method and compared to traditional methods.

VI. DISCUSSION

The algorithm for unsupervised classification presented here is based on a mixture model using ICA to model the structure of the classes. The parameters are estimated using maximum likelihood. This method is similar to other approaches including the mixture density networks by Bishop [6] in which a neural network was used to find arbitrary density functions. Our algorithm reduces to the Gaussian mixture model when the source priors are Gaussian. Purely Gaussian structure, however, is rare in real data sets. Here we have used super-Gaussian and sub-Gaussian densities as priors. These priors could be extended as proposed by Attias [1]. Our model learned a complete set of basis functions without additive noise. However, the method can be extended to take into account additive Gaussian noise and an over-complete set of basis vectors [31], [33]. The structure of the ICA mixture model is also similar to the mixtures of factor analyzers proposed by Ghahramani and Hinton [19]. The difference here

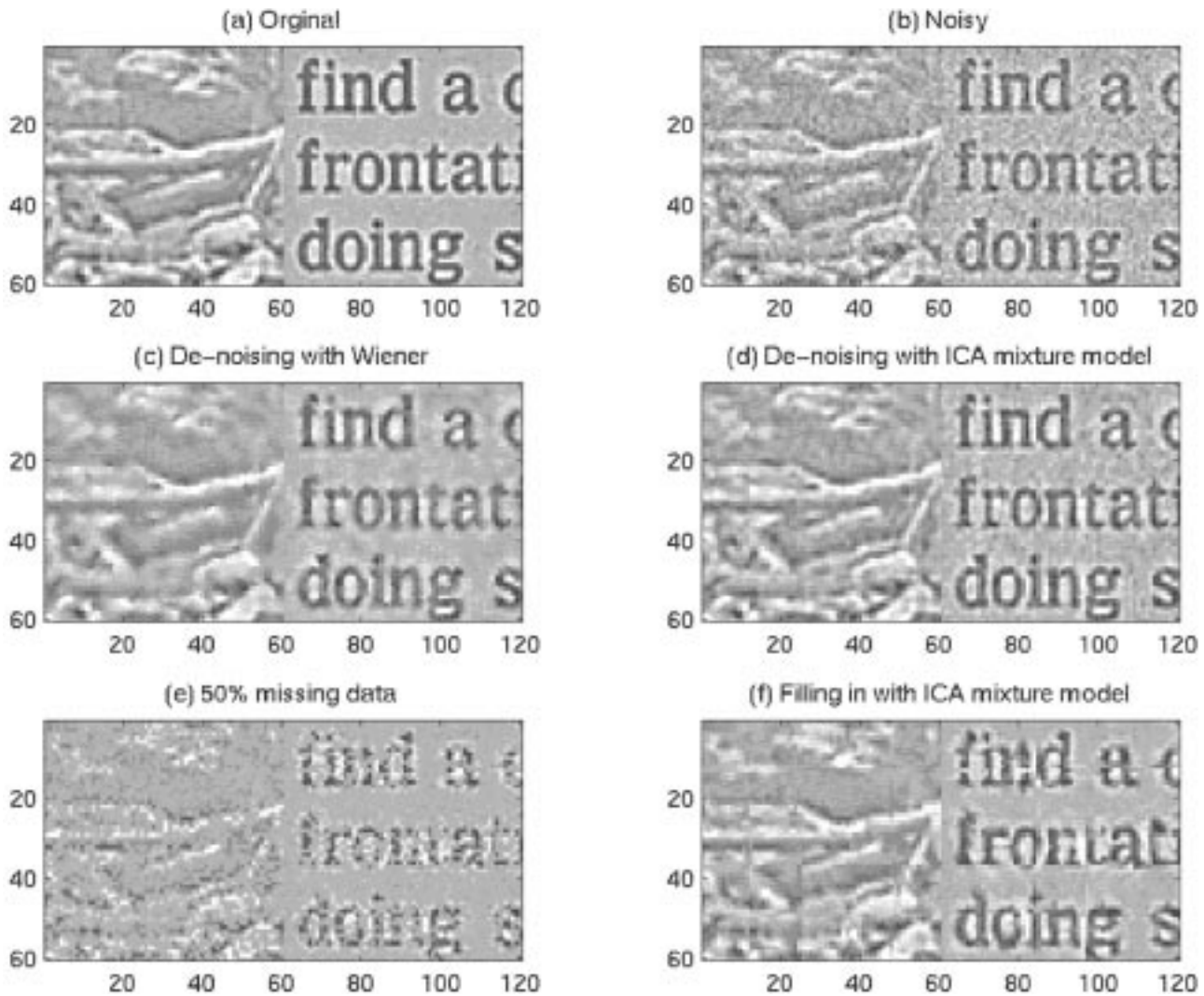


Fig. 6. (a) Original image, (b) noisy image (SNR = 13 dB), (c) results of the Wiener filtering denoising method (SNR = 15 dB), and (d) reconstructed image using the ICA mixture model (SNR = 21 dB), (e) image with 50% missing pixels replaced with gray pixels (SNR = 7 dB), and (f) reconstructed image using the ICA mixture model (SNR = 14 dB). The reconstruction by interpolating with splines was 11 dB.

is that the coefficient distribution $p(s)$ and hence the distribution $p(\mathbf{X}|\Theta)$ are assumed to be non-Gaussian. Another extension of this into modeling temporal information by Penny *et al.* [37] could be considered.

Note that the ICA mixture model is a nonlinear model in which the data structure within each class is modeled using linear superposition of basis functions. The choice of class, however, is nonlinear because the classes are assumed to be mutually exclusive. This model is therefore a type of nonlinear ICA model and it is one way of relaxing the independence assumption over the entire data set. The ICA mixture model is a conditional independence model, i.e., the independence assumption holds within only each class and there may be dependencies among the classes. A different view of the ICA mixture model is to think of the classes as an overcomplete representation. Compared to the approach in [31] and [33], the main difference is that the basis functions learned here are mutually exclusive, i.e., each class used its own (complete) set of basis functions.

We have demonstrated that the algorithm can learn efficient codes to represent different image types such as natural scenes and text images and was a significant improvement over PCA encoding. Single class ICA models showed image compression rates comparable to or better than traditional image compression algorithms such as JPEG [32]. Using ICA mixtures to learn image codes should yield additional improvement in coding efficiency [29]. Here, we have investigated the application of the ICA mixture model to the problem of unsupervised classification and segmentation of images as well as denoising, and filling-in missing pixels. Our results suggest that the method is capable of handling the problems successfully. Furthermore, the ICA mixture model is able to increase the performance over Gaussian mixture models or standard ICA models when a variety of image types are present in the data. In terms of increased image compression performance we believe that switching between complete set of basis functions where each set is optimally tuned toward the data will improve the compression ratio for those trained specific data

sets. Our future work will consider image compression results on benchmark data compared with current image coding systems.

The unsupervised image segmentation results suggest that our method can be used as a baseline method but for good performance we need additional methods to cover the global structure of the images. Therefore, unsupervised image segmentation by discovering basis functions of image textures only cannot be the sole solution to this difficult problem. Since the segmentation technique presented here is based on the classification of small image patches, the global information of the image is not taken into consideration. The multiresolution problem may be overcome by including a multiscale hierarchical structure into the algorithm or by reapplying the algorithm with different scales of the basis functions and combining the results. This additional process would smooth the image segmentation and the ICA mixture model could serve as a baseline segmentation algorithm. These results need to be compared with other methods, such as those proposed in [15] which measured statistical properties of textures coded with a large-scale, fixed-wavelet basis. In contrast, the approach here models image structure by adapting the basis functions themselves in a chosen scale. It can therefore serve as a baseline classification algorithm and then be extended with traditional segmentation techniques that take into account the global image information. We emphasize that our method is based on density estimation and not texture synthesis. The latter may produce good looking textures but may not be useful for classification purposes which depends more on how well the statistical structure of the images is described in [17]. Other related works for image segmentation are described in [35].

The application of ICA for noise removal in images as well as filling in missing pixels will result in significant improvement when several different classes of images are present in the image. Fax machines for example transmit text as well as images. Since the basis functions of the two image models are significantly different [28] the ICA mixture model will improve in coding and enhancing the images. The technique used here for denoising and filling-in missing pixels was proposed in [32] and [33]. The same technique can be applied to multiple classes as demonstrated in this paper. The main concern of this technique is the accuracy of the coefficient prior. A different technique for denoising using the fixed point ICA algorithm was proposed in [22] which may be intuitively sound but requires some tweaking of the parameters. Other related works for image enhancement is described in [13].

Another issue not addressed in this paper is the relevance of the learned codes to neuroscience. The principle of redundancy reduction for neural codes is preserved by this model and some properties of V1 receptive fields are consistent with recent observations [5], [32], [36]. It is possible that the visual cortex uses overcomplete basis sets for representing images; this raises the issue of whether there are cortical mechanisms that would allow switching to occur between these bases depending on the input.

In conclusion, the ICA mixture model has the advantage that the basis functions of several image types can be learned simultaneously. Compared with algorithms that use one fixed set

ICA Mixture Model

```

Initialize model parameters  $\theta_1, \dots, \theta_K$ 

Input data vectors  $\mathbf{x}_1, \dots, \mathbf{x}_T$ 

Repeat

    Perform main adaptation loop: adapt class parameters

    Adapt the class probability for each class

    Optional: Adapt the number of classes

Until the adaptation has converged

Assign each data vector  $\mathbf{x}_t$  to one of the classes

```

Fig. 7. Unsupervised adaptation and classification using the ICA mixture model: Adapt class parameters, classify, and separate data.

Main Adaptation Loop

```

Initialize data index  $t=1$ 

Repeat

    For each class,

        calculate  $\mathbf{s}_k, p(\mathbf{s}_k)$  and  $p(\mathbf{x}|\Theta)$ 

        calculate  $p(C_k|\mathbf{x}_t, \Theta)$ 

        adapt the mixing matrix  $\mathbf{A}_k$ 

        adapt the bias vector  $\mathbf{b}_k$ 

        adapt the pdf parameters for the sources

    End

Until  $t=T$ 

```

Fig. 8. Perform main adaptation loop: adapt the class parameters Θ for all classes and all data vectors given the previously computed parameters.

of basis functions, the results presented here are promising and may provide further insights in designing improved image processing systems.

APPENDIX

IMPLEMENTATION OF THE ICA MIXTURE MODEL

Figs. 7 and 8 summarize the adaptation or training procedure for the ICA mixture model. First, the all model parameters are initialized for all classes. Second, data is sampled usually in random order so that there is no correlation in time (unless it is desired to have model correlation over time). The main part of the algorithm is the adaptation of the class parameters, the class probability adaptation and the estimation of the number of classes. The latter estimation is optional and described in [2] since the number of classes may be known in the application. The class probability adaptation is averaging over the number of data vectors belonging to that class and is essentially an average

De-noising procedure

```

Resize image into small patches

Initialize data index t=1

Repeat
  For each class,
    calculate  $\mathbf{s}_k$ 
    calculate  $p(\mathbf{s}_k)$  and  $p(\mathbf{x}|\Theta)$ 
    calculate  $p(C_k|\mathbf{x}_t, \Theta)$ 
    MAP estimate for  $\hat{\mathbf{s}}_k$ 
    compute  $\hat{\mathbf{x}}_t$ 
  End
  pick  $\hat{\mathbf{x}}_t$  of highest class probability
Until t=T

```

Fig. 9. Denoising procedure.

of (5) for each class. The adaptation loop is repeated until convergence is achieved where convergence is measured in terms of maximum likelihood. Practically, the adaptation is stopped once the log-likelihood function stabilizes asymptotically with increasing number of iterations. Note that this converges speeds up significantly when new data is sampled randomly after each loop.

Fig. 8 summarizes the main adaptation or training for the class parameters. For each class, the source vectors are computed, the probability of the source vector is estimated as in (10) or (11), which in turn can be used to compute the likelihood function in (4). Given these computations, the class probability can be computed for all classes as in (5). The adaptation of the class parameters are: the mixing matrix \mathbf{A}_k as in (6) or more specific as in (8) and (11), the bias vector \mathbf{b}_k as in (7), and the pdf parameters for each source as in (9) or the exponential power function in [27].

Fig. 9 summarizes the main steps in denoising image patches. An image is divided into small patches. For each patch and for all classes, the source vectors, the source probability, the data likelihood, the class probability, the MAP estimate for the sources and the projection onto the denoised data vector are calculated given all the model parameters. The denoised image patch with highest class probability given the models and the noisy image patch is chosen as the denoised image patch.

ACKNOWLEDGMENT

The authors thank the anonymous reviewers for their detailed comments and questions which improved the quality of the presentation of this paper.

REFERENCES

- [1] H. Attias, "Blind separation of noisy mixtures: An EM algorithm for independent factor analysis," *Neural Comput.*, vol. 11, pp. 803–851, 1999.
- [2] U.-M. Bae, T.-W. Lee, and S.-Y. Lee, "Blind signal separation in teleconferencing using the ica mixture model," *Electron. Lett.*, vol. 36, no. 7, pp. 680–682, 2000.
- [3] H. Barlow, *Sensory Communication*. Cambridge, MA: MIT Press, 1961, pp. 217–234.
- [4] A. J. Bell and T. J. Sejnowski, "An information-maximization approach to blind separation and blind deconvolution," *Neural Comput.*, vol. 7, pp. 1129–1159, 1995.
- [5] —, "The 'independent components' of natural scenes are edge filters," *Vis. Res.*, vol. 37, no. 23, pp. 3327–3338, 1997.
- [6] C. M. Bishop, "Mixture density networks," Tech. Rep. NCRG/4288, 1994.
- [7] A. Blake and A. Zisserman, *Visual Reconstruction*. Cambridge, MA: MIT Press, 1987.
- [8] L. Bottou, P. Haffner, P. Howard, P. Simard, Y. Bengio, and Y. LeCun, "High quality document image compression with DJVU," *J. Electron. Imag.*, 1998.
- [9] G. Box and G. Tiao, *Bayesian Inference in Statistical Analysis*. New York: Wiley, 1973.
- [10] R. W. Buccigrossi and E. P. Simoncelli, "Image compression via joint statistical characterization in the wavelet domain," *IEEE Trans. Image Processing*, vol. 8, pp. 1688–1701, Dec. 1999.
- [11] J.-F. Cardoso and B. Laheld, "Equivariant adaptive source separation," *IEEE Trans. Signal Processing*, vol. 45, pp. 434–444, Feb. 1996.
- [12] J. F. Cardoso, "High-order contrasts for independent component analysis," *Neural Comput.*, vol. 11, pp. 157–192, 1999.
- [13] T. Chan and J. Shen. (2000, Mar.) Mathematical models for local deterministic inpaintings. [Online]. Available: www.math.ucla.edu
- [14] P. Comon, "Independent component analysis—A new concept?," *Signal Process.*, vol. 36, no. 3, pp. 287–314, 1994.
- [15] J. S. De Bonet and P. Viola, "A nonparametric multi-scale statistical model for natural images," in *Proc. Advances in Neural Information Processing Systems*, 1997.
- [16] R. O. Duda and P. E. Hart, *Pattern Classification and Scene Analysis*. New York: Wiley, 1973.
- [17] A. Efros and T. Leung, "Texture synthesis by nonparametric sampling," in *Int. Conf. Computer Vision*, Corfu, Greece, Sept. 1999.
- [18] D. Field, "What is the goal of sensory coding?," *Neural Comput.*, vol. 6, pp. 559–601, 1994.
- [19] Z. Ghahramani and G. E. Hinton, "The EM algorithm for mixtures of factor analyzers," Dept. Comput. Sci., Univ. Toronto, Toronto, ON, Canada, Tech. Rep., 1997.
- [20] M. Girolami, "An alternative perspective on adaptive independent component analysis algorithms," *Neural Comput.*, vol. 10, no. 8, pp. 2103–2114, 1998.
- [21] A. Hyvaerinen and E. Oja, "A fast fixed-point algorithm for independent component analysis," *Neural Comput.*, vol. 9, pp. 1483–1492, 1997.
- [22] P. Hyvaerinen, A. Hoyer, and E. Oja, "Sparse code shrinkage: Denoising by nonlinear maximum likelihood estimation," in *Proc. Advances in Neural Information Processing Systems*, 1999.
- [23] C. Jutten and J. Herault, "Blind separation of sources, part I: An adaptive algorithm based on neuromimetic architecture," *Signal Process.*, vol. 24, pp. 1–10, 1991.
- [24] T.-W. Lee, K. Chan, and M. Lewicki, "The generalized Gaussian mixture model," INC Tech. Rep., Univ. California, San Diego, 2001.
- [25] T.-W. Lee, M. Girolami, A. J. Bell, and T. J. Sejnowski, "A unifying framework for independent component analysis," *Comput. Math. Applicat.*, vol. 39, no. 11, pp. 1–21, Mar. 2000.
- [26] T.-W. Lee, M. Girolami, and T. J. Sejnowski, "Independent component analysis using an extended infomax algorithm for mixed sub-Gaussian and super-Gaussian sources," *Neural Comput.*, vol. 11, no. 2, pp. 409–433, 1999.
- [27] T.-W. Lee and M. S. Lewicki, "The generalized Gaussian mixture model using ICA," in *Proc. Int. Workshop ICA*, vol. 239–244, June 19–22, 2000.
- [28] T.-W. Lee, M. S. Lewicki, and T. J. Sejnowski, "Unsupervised classification with non-Gaussian mixture models using ICA," in *Advances in Neural Information Processing Systems*. Cambridge, MA: MIT Press, 1999, pp. 508–514.

- [29] —, “Unsupervised classification with non-Gaussian sources and automatic context switching in blind signal separation,” *Pattern Recognit. Mach. Learn.*, vol. 22, pp. 1–12, Oct. 2000.
- [30] M. Lewicki and B. Olshausen, “Inferring sparse, overcomplete image codes using an efficient coding framework,” in *Proc. Advances in Neural Information Processing Systems*, 1998, pp. 556–562.
- [31] M. Lewicki and T. J. Sejnowski, “Learning nonlinear overcomplete representations for efficient coding,” in *Proc. Advances in Neural Information Processing Systems*, 1998, pp. 815–821.
- [32] M. S. Lewicki and B. A. Olshausen, “A probabilistic framework for the adaptation and comparison of image codes,” *J. Opt. Soc. Amer. A*, vol. 16, no. 7, pp. 1587–1601, 1999.
- [33] M. S. Lewicki and T. J. Sejnowski, “Learning overcomplete representations,” *Neural Comput.*, vol. 12, no. 2, pp. 337–365, 2000.
- [34] M. S. Lewicki, “A flexible prior for independent component analysis,” Univ. California, San Diego, INC Tech. Rep., 2000.
- [35] B. S. Manjunath, G. M. Haley, and W. Y. Ma, “Multi-band approaches for texture classification and segmentation,” in *Handbook of Image and Video Processing*, A. Bovik, Ed. New York: Academic, 2000, pp. 367–381.
- [36] B. Olshausen and D. Field, “Emergence of simple-cell receptive field properties by learning a sparse code for natural images,” *Nature*, vol. 381, pp. 607–609, 1996.
- [37] W. Penny, R. Everson, and S. Roberts, *Advances in Independent Components Analysis*, M. Girolami, Ed. Berlin, Germany: Springer-Verlag, 2000, pp. 3–22.
- [38] J. Stutz and P. Cheeseman, “Autoclass—A bayesian approach to classification,” in *Maximum Entropy and Bayesian Methods*. Norwell, MA: Kluwer, 1994.
- [39] S. C. Zhu, Y. N. Wu, and D. Mumford, “Minimax entropy principle and its application to texture modeling,” *Neural Comput.*, vol. 9, no. 8, pp. 1627–1660, 1997.



Te-Won Lee was born in Chungnam, Korea, in 1969. He received the diploma degree in March 1995 and the Ph.D. degree (with highest honors) in electrical engineering in October 1997 from the University of Technology, Berlin, Germany.

He was a Visiting Graduate Student at the Institute Nationale Polytechnique de Grenoble, France, the University of California at Berkeley, and Carnegie Mellon University, Pittsburgh, PA. From 1995 to 1997, he was a Fellow with Max-Planck Institute, Garching, Germany. He is currently a Research

Professor with the Institute for Neural Computation, University of California at San Diego (UCSD), La Jolla, and a Research Associate with The Salk Institute, UCSD. His research interests are in the areas of unsupervised learning algorithms, artificial neural networks and Bayesian probability theory with applications to signal and image processing.



Michael S. Lewicki received the B.S. degree in mathematics and cognitive science in 1989 from Carnegie Mellon University (CMU), Pittsburgh, PA, and the Ph.D. degree in computation and neural systems from the California Institute of Technology, Pasadena, in 1996.

From 1996 to 1998, he was a Postdoctoral Fellow with the Computational Neurobiology Laboratory at the Salk Institute, University of California at San Diego, La Jolla. He is currently Assistant Professor with the Computer Science Department, CMU, and in the CMU/University of Pittsburgh Center for the Neural Basis of Cognition. His research involves the study and development of computational approaches to the representation, processing, and learning of structure in natural patterns.