Capturing Nonlinear Dependencies in Natural Images using ICA and Mixture of Laplacian Distribution

Hyun-Jin Park and Te-Won Lee
Institute for Neural Computation (INC), University of California San Diego (UCSD),
9500 Gilman Drive, La Jolla, CA 92093-0523
{hjinpark, tewon}@ucsd.edu

Abstract:

Capturing dependencies in images in an unsupervised manner is important for many image-processing applications and for understanding the structure of natural image signals. Data generative linear models such as principal component analysis (PCA) and independent component analysis (ICA) have shown to capture low level features such as oriented edges in images. However those models capture only linear dependency and therefore its modeling capability is limited. We propose a new method for capturing nonlinear dependencies in images of natural scenes. This method is an extension of the linear ICA method and builds on a hierarchical representation. The model makes use of lower level linear ICA representation and a subsequent mixture of Laplacian distribution for learning the nonlinear dependencies in an image. The model parameters are learned via the expectation maximization (EM) algorithm and it can accurately capture variance correlation and other high order structures in a simple and consistent manner. We visualize the learned variance correlation structure and demonstrate applications to automatic image segmentation and image denoising.

Keywords: ICA, nonlinear, dependency, EM algorithm, image segmentation, denoising

1. Introduction

Unsupervised learning has become an important tool for understanding biological information processing and developing intelligent signal processing algorithms [1]. Successful adaptive learning algorithms for signal and image processing have been developed recently, but their capabilities are far from performances of real biological systems that are more flexible and robust. One of the main reasons for this deficiency is due to the lack of an efficient signal representation that is usually learned through experience and data. Many methods available can capture low level signal structure, e.g. sharp edges in images. But most method lack in a description of the image in terms of its higher-level structure including object like structures, textures and certain low level invariances. Humans are good at capturing those high level structures and require little prior knowledge or supervision. Learning algorithms that capture sophisticated representations in an unsupervised manner can provide a better understanding of neural information processing and also provide novel learning algorithms for signal processing applications.

Recently adaptive techniques have gained popularity due many potential applications and its use in analyzing data. Independent component analysis (ICA) for example has been effective at learning representations for images of natural scenes that have similar properties than receptive fields in the visual cortex. Those learned features can be applied for feature extraction, denoising and image segmentation tasks [1][2][3][4][5][6]. Although the initial results are interesting, this method is limited and restrictive due to the linear model assumption. This is because the ICA algorithm is constrained to capturing linear dependency only. The learned filters for image representation for example show low level features that capture localized oriented edges in an image. Images however, have other types of dependencies across the entire image. This is in particular evident in images that display different texture patterns. ICA may be able to capture certain edges in a texture but the ICA does not capture the texture information.

We are interested in learning those dependencies that are more global across the entire image. In a sense this is similar to learning textures in an image. However, we are interested in learning classes of textures not necessarily for the purpose of image segmentation but more for the purpose of encoding the image in an efficient manner. The segmentation of the image is a mere side product of this learning process. Another viewpoint and motivation for learning those dependencies is the desire to find an object-like representation.

There are several approaches that have been proposed to learn nonlinear dependencies. We first describe the basic ICA model and nonlinear ICA extensions for image processing.

1.1 Basic ICA model

Barlow argued that biological information processing systems could be self-organized based on statistical property of the input signal [1][7]. He argued that the goal of the visual system is to reduce the information redundancy among the input sensory information. Field (1994) suggested a compact representation using PCA [8][9] and Olshausen and Field (1996) proposed a neural network with explicit sparseness constraint to learn image basis functions [10][11]. When the ICA learning rule is applied to image patches of natural scenes, the adapted basis functions are localized and orientation sensitive, which are similar to properties of simple cell receptive fields of biological visual system [1][8][9][12][13]. In ICA model, statistics of natural image patches are approximated well by Eq.(1~2) [14][15][16][17], where X is the measured data vector and u is a source signal vector whose distribution is a product of sparse distributions such as generalized Laplace distributions [1][3][18]. A is a matrix that defines linear mapping from u to X.

$$X = Au \tag{1}$$

$$P(u) = \prod_{i}^{M} P(u_i) \tag{2}$$

ICA learns linear dependency between pixels [14][15][16]. It finds strongly correlated axis and removes linear correlation in projected encoding signals as far as a linear model can do. But ICA captures only linear dependency and *its modeling capability is limited* up to this linear dependency.

The representations learned by *ICA algorithm are relatively low-level representations*. There are more high-level representations in biological systems that are correlated with contours, textures, and objects. By learning such high-level representations, we can develop signal-processing algorithms that exhibit similar properties to biological intelligence. In previous approaches, it is shown that by capturing nonlinear dependencies beyond linear, one can learn such higher-level representations similar to biological systems [19][20][21][22].

Studies beyond linear ICA model are focused on *variance correlation of source signal* after ICA processing. In linear ICA model, source signals are assumed to be independent from each other. But given natural image signals and ICA model, the variances of source signals tend to be correlated. Although it is difficult to linearly predict a source signal from each other, but their variances can be predicted from others [23][24]. So with the variance dependency, a signal is still ‘predictable’ in a nonlinear manner. This higher-order dependency cannot be captured by conventional ICA model, and we need more sophisticated model.

1.2 Learning nonlinear dependencies beyond ICA model

Modeling variance dependencies beyond linear model is a key to understanding biological signal processing and developing new signal-processing algorithms. And there are several previous approaches to model variance dependency beyond the ICA model [19][20][21][22][25]. Their results showed that one could learn more high-level representations similar to biological systems by considering nonlinear variance dependency.

Hyvarinen and Hoyer suggested to use a special variance related distribution function to model the variance correlation [19][20][21]. His model uses distribution of Eq. (3) to model two dependent signals. The conditional distribution of this distribution shows variance dependency similar to that one found in natural image signals. In Eq. (3), u_1 and u_2 are 1-D random variables and c is a constant.

$$P(u_1, u_2) = c\exp\left(-\sqrt{u_1^2 + u_2^2}\right) \tag{3}$$

This model can learn grouping of dependent sources (Subspace ICA) or topographic arrangements of correlated sources (Topographic ICA) [19][20]. The learned topographic structures show high similarity to the topographical arrangements of simple cells in the visual cortex, which says that the neural information processing system seems to be organized to reflect the nonlinear dependency structure of visual signals [21].

But the learned subspace or topographical arrangements does not provide high level structures explicitly. The subspace model [19] only provides grouping of subspaces. And the topographic model is limited to providing only neighborhood relationship [20]. It is not clear how to compute the higher order signal encodings and its capability to signal processing application is limited.

Portilla et al. used Gaussian Scale Mixture (GSM) to model variance dependency in wavelet domain. This model can learn variance correlation in source prior and showed improvement in image denoising [26]. But in this model, dependency is defined only between a subset of wavelet coefficients. Similarly, Welling et al. suggested a product of expert model where each expert represents a variance correlated group [25]. The product form of the model enables applications to image denoising. But these models don't reveal higher-order structures explicitly.

Our model is motivated by a recent work by Lewicki and Karklin. Lewicki and Karklin proposed a hierarchical two-stage model where first stage is an ICA model and second-stage source signal v generates variances for ICA source signal u as shown in Eq. (4~5) [22]. In the equation, u, v and λ are vectors and B is a matrix consisting of variance basis. They introduced a second stage generative model that linearly generates variances for the first stage (ICA) source signals. The first layer source signals are assumed to be independent given the variance signal generated from the second layer.

$$P(u \mid \lambda) = c \exp\left(-\left| \frac{u}{\lambda} \right|^{q} \right) \tag{4}$$

$$\log[\lambda] = Bv \tag{5}$$

The most important contribution of their work is that it explicitly learns variance correlations in a form of generating basis. In this model can learn higher order structures or 'variance basis' encoded in B. Those variance bases can be interpreted as textures or building blocks in an image.

But in Lewicki's model, treating variance as another random variable introduces a high complexity and rough approximation [22], thus makes it difficult to derive clear parametric distribution model of ICA source signal u. It is therefore difficult to apply this model for practical image processing applications.

In summary, there are limitations in previous approaches to capture nonlinear dependency. Hyvarinen's model [19][20][21] represents nonlinear dependency only indirectly. Lewicki's model [22] provides a direct representation but introduces high complexity and approximation in the model. So their *application to practical signal processing problems is limited.* A simple parametric model of nonlinearly correlated signals is necessary for a better understanding of natural image signals and building intelligent systems.

In this paper, we propose a parametric mixture model that can learn nearly independent variance correlated source signals. Our model can be considered as a simplification of model in [22] by constraining v to be 0/1 random vector where only one element can be 1. The proposed model allows direct representation of variance correlation structure and reveals high order structure of the image signals. Also our model provides a simple parametric distribution model for ICA source signals and can be used for better signal processing such as denoising.

We derive the parametric mixture model and its learning algorithm in Section 2. Then we visualize learned variance structure for each mixture in Section 3, which shows high-level structures. We compare of learned model with the real signal histogram. Finally, we demonstrate the application of the learned model to image segmentation and denoising.

2. Learning nonlinear dependencies by Laplacian Mixture Model

We propose a hierarchical 2-stage model where the first stage is an ICA model and the second stage is a mixture model that captures variance correlated source prior (figure 1). The correlation of variance in natural images reflects different types of regularities in the real world. Such specialized regularities can be called as 'contexts'. Different contexts give rise to different variances. To model such context dependent variance correlation, we use mixture models where each mixture is a Laplacian distribution with 0-mean and different variances. Different Laplacian distributions model different contexts. The ICA matrix A is learned in prior independently of the second stage. Then the second stage is trained given the first stage ICA source signals.

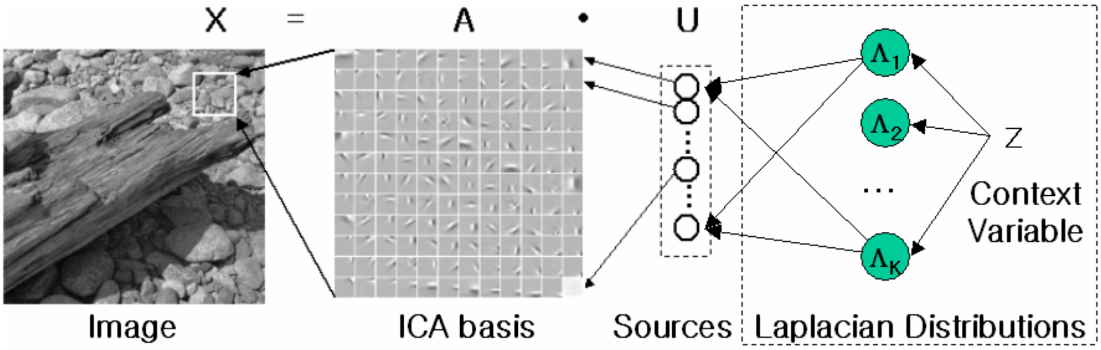


Figure 1. Proposed mixture model for learning higher order structure. ICA source signal U is modeled by a mixture model with different variance Λ_k. Z is a discrete random vector that represents which mixture is responsible for the given image patch. Each multi-dimensional Laplacian distribution has 0-mean and different variances.

In figure 1, a context variable Z 'selects' which Laplacian distribution will represent ICA source signal u for each image patch. And each Laplacian distribution is a factorial multi-dimensional distribution. *The advantage of Laplacian distribution for modeling context is that we can model a sparse distribution using only one Laplacian distribution. But we need more than two Gaussian distributions to do the same thing.* Also conventional ICA is approximated well as a special case of our model with single Laplacian.

Each Laplacian distribution represents a specific visual context, where some dimensions have high variances while the others have low variance. Each Laplacian distribution is factorial but as a whole, the mixture model is not fully factorial as the conventional ICA source model. It can learn nearly independent variance-correlated signal distribution. The detailed model definition and its learning algorithm will be shown in next sections.

ICA Mixture models have been used for image segmentation and denoising [5][6]. But in previous approaches each mixture has different mean, and it doesn't allow two mixtures with different variances to overlap. Thus previous mixture approaches doesn't capture variance correlation at all. In our model, each Laplacian distribution has same 0-mean and is highly overlapped with each other around origin.

The parametric PDF provided by our model is an important advantage for designing adaptive signal processing applications. Utilizing this advantage, we demonstrate simple applications on image segmentation and denoising. In image segmentation, we ‘identify’ the hidden context variable Z for each image patch, detecting high-level structures of natural images. In image denoising, we can think of our model as a better source prior of ICA model and can apply the learned source prior in Bayesian MAP (maximum a posteriori) estimation problem. In summary segmentation based on our model shows clear correlation with the high level structures of the scene. In Bayesian MAP denoising our model shows substantial improvements over conventional denoising methods.

2.1 Mixture of Laplacian Distribution Model

The first stage of our model is the conventional ICA model. We define the mixture model for the second stage and derive its learning algorithm here. The key idea of our model is that the distribution of ICA source signal u can be modeled as mixture of Laplace distributions with specialized variances.

The PDF for a 1-D Laplacian distribution can be written as Eq. (6), where u is a 1-D random variable, and λ is a 1-D variance parameter (λ>0).

$$P(u \mid \lambda) = \frac{1}{(2\lambda)} \exp\left(-\left| \frac{u}{\lambda} \right| \right) \tag{6}$$

This can be extended to a factorial M-dimensional distribution as shown in Eq. (7)

$$P(\vec{u}_n \mid \vec{\lambda}_k) = \prod_m^M \frac{1}{(2\lambda_{k,m})} \exp\left(-\frac{|u_{n,m}|}{\lambda_{k,m}} \right) \tag{7}$$

Where $\vec{u}_n = (u_{n,1}, u_{n,2},,, u_{n,M})$ is a M-dimensional random vector of n-th data, $\vec{\lambda}_k = (\lambda_{k,1}, \lambda_{k,2}, ..., \lambda_{k,M})$ is a M-dimensional variance vector of k-th Laplacian distribution (different Laplacian distributions are indexed by k), and $\lambda_{k,m} > 0$ for all k, m.

If we consider a mixture of Laplace distributions, we need to introduce mixture probability Π in addition to variance parameter Λ. Then we can write the likelihood of data U given Π and Λ as below.

$$\begin{aligned} P(U \mid \Lambda, \Pi) &= \prod_n^N P(\vec{u}_n \mid \Lambda, \Pi) = \prod_n^N \sum_k^K \pi_k P(\vec{u}_n \mid \vec{\lambda}_k) \\ &= \prod_n^N \sum_k^K \pi_k \prod_m^M \frac{1}{(2\lambda_{k,m})} \exp\left(-\frac{|u_{n,m}|}{\lambda_{k,m}} \right) \end{aligned} \tag{8}$$

Where N : number of data samples (here, data means ICA source signal u).
K : number of mixtures (Laplacian distributions)
M : dimension of data samples (same as the dimension of Laplacian distributions)
$u_n = (u_{n1},,, u_{nm},,, u_{nM})$: n-th data sample
$\lambda_k = (\lambda_{k1},,, \lambda_{km},,, \lambda_{kM})$: Variance of k-th Laplacian distribution.
π_k : probability of mixture k s.t. $\sum_k \pi_k = 1$
$U = (\vec{u}_1, \vec{u}_2,,, \vec{u}_i,,, \vec{u}_N)$: Ensemble of all data samples

$\Lambda = (\vec{\lambda}_1, \vec{\lambda}_2,,, \vec{\lambda}_k,,, \vec{\lambda}_K)$: Ensemble of all variance vectors.

$\Pi = (\pi_1,,, \pi_K)$: Ensemble of all mixture probability

In general, we can compute ML (maximum likelihood) or MAP (maximum a posteriori) estimation of the parameters Π and Λ given the data. But the optimization is difficult because of the summation part and we need to apply EM (expectation maximization) algorithm for learning [27][28].

To apply the EM method, we need to introduce a hidden variable Z, that determines which mixture component k, is responsible for a given data sample. Once we assume the hidden variable Z is known, we can write the likelihood of data and hidden variable as Eq. (9).

$$P(U, Z \mid \Lambda, \Pi) = \prod_{n}^{N} P(\vec{u}_n, Z \mid \Lambda, \Pi)$$
$$= \prod_{n}^{N} \left[\prod_{k}^{K} \left[(\pi_k)^{z_k^n} \prod_{m} \left(\left(\frac{1}{2\lambda_{k,m}} \right)^{z_k^n} \cdot \exp\left(- z_k^n \frac{|u_{n,m}|}{\lambda_{k,m}} \right) \right) \right] \right] \tag{9}$$

Where z_k^n : Latent random variables

(z_k^n is 1 if n-th data sample is generated from k-th mixture, 0 otherwise)

$Z = \left(z_k^n \right)$: Ensemble of z_k^n (a N by K matrix), and $\sum_k z_k^n = 1$ for all n = 1…N.

The constraint $\sum_k z_k^n = 1$ is clear from the definition. For any data sample n, we know that it should have been generated from only one of the mixtures. So given a data n, there is one and only one k such that $z_k^n = 1$ and $z_{k'}^n = 0$ $(k \neq k')$.

2.2 EM algorithm for parameter learning

EM algorithm works by maximizing the log likelihood of data, averaged over hidden variable Z [27][28]. It works by maximizing the Expectation of the log likelihood over hidden variable. The log likelihood of data given parameters can be derived from Eq. (9) as below.

$$\log P(U, Z \mid \Lambda, \Pi) = \sum_{n,k} \left[z_k^n \log(\pi_k) + \sum_m z_k^n \left(\log(\frac{1}{2\lambda_{k,m}}) - \left| \frac{u_{n,m}}{\lambda_{k,m}} \right| \right) \right] \tag{10}$$

Then the expectation can be written as Eq. (11).

$$E\{\log P(U, Z \mid \Lambda, \Pi)\} = \sum_{n,k} E\{z_k^n \mid U, \Lambda, \Pi\} \left[\log(\pi_k) + \sum_m \left(\log(\frac{1}{2\lambda_{k,m}}) - \left| \frac{u_{n,m}}{\lambda_{k,m}} \right| \right) \right] \tag{11}$$

The expectation can be evaluated, only if we are given the data U and some estimated parameters of Λ and Π. We can use a temporary estimation Λ' and Π' for Λ and Π in EM method. From here, we abbreviate $E\{z_k^n \mid U, \Lambda', \Pi'\}$ as $E\{z_k^n\}$.

$$
\begin{aligned}
&E\{z_k^n\} \equiv E\{z_k^n \mid U, \Lambda', \Pi'\} \\
&= \sum_{z_k^n=0}^{1} z_k^n P(z_k^n \mid u_n, \Lambda', \Pi') = P(z_k^n = 1 \mid u_n, \Lambda', \Pi') \\
&= \frac{P(u_n \mid z_k^n = 1, \Lambda', \Pi') P(z_k^n = 1 \mid \Lambda', \Pi')}{P(u_n \mid \Lambda', \Pi')} \\
&= \frac{\left(\prod_{m}^{M} \frac{1}{2\lambda_{k,m}'} \exp\left(-\frac{|u_{n,m}|}{\lambda_{k,m}'}\right)\right) \cdot \pi_k'}{P(u_n \mid \Lambda', \Pi')} = \frac{1}{c_n} \prod_{m}^{M} \frac{\pi_k'}{2\lambda_{k,m}'} \exp\left(-\frac{|u_{n,m}|}{\lambda_{k,m}'}\right)
\end{aligned}
\tag{12}
$$

Then the normalization constant for a given sample n can be computed by summation over k.

$$
\begin{aligned}
&c_n = P(u_n \mid \Lambda', \Pi') = \sum_{k}^{K} P(u_n \mid z_k^n = 1, \Lambda', \Pi') P(z_k^n = 1 \mid \Lambda', \Pi') \\
&= \sum_{k=1}^{K} \pi_k \prod_{m=1}^{M} \frac{1}{(2\lambda_{k,m}')} \exp\left(-\left|\frac{u_{n,m}}{\lambda_{k,m}'}\right|\right)
\end{aligned}
\tag{13}
$$

The EM algorithm works by maximizing equation (11), given the expectation $E\{z_k^n\}$ computed from Eq. (12). Those expectations can be computed using data U and parameters Λ' and Π' estimated in previous iteration of EM algorithm. These correspond to the Expectation step of EM algorithm.

In the next step (maximization step) of EM algorithm, we need to maximize the expected log likelihood over parameter Λ and Π.

2.2.1 Maximization over parameter Λ

The gradient of log likelihood in equation (11) with respect to parameter Λ can be computed as Eq. (14), where we utilize the fact that $\frac{\partial \lambda_{k',m'}}{\partial \lambda_{k,m}} = 0$ for $k' \neq k, m' \neq m$.

$$
\begin{aligned}
&\frac{\partial E\{\log P(U, Z \mid \Lambda, \Pi)\}}{\partial \lambda_{k,m}} \\
&= \frac{\partial \sum_{n,k'} E\{z_{k'}^n\} \left[\log(\pi_{k'}) + \sum_{m'} \left(\log(\frac{1}{2\lambda_{k',m'}}) - \frac{|u_{n,m'}|}{\lambda_{k',m'}}\right)\right]}{\partial \lambda_{k,m}} \\
&= \sum_{n,k'} E\{z_{k'}^n\} \left[\frac{\partial}{\partial \lambda_{k,m}} \sum_{m'} \left(\log(\frac{1}{2\lambda_{k',m'}}) - \frac{|u_{n,m'}|}{\lambda_{k',m'}}\right)\right] \\
&= \sum_{n,k'} E\{z_{k'}^n\} \left[\sum_{m'} \left(-\frac{\partial}{\partial \lambda_{k,m}} \log(2\lambda_{k',m'}) - \frac{\partial}{\partial \lambda_{k,m}} \frac{|u_{n,m'}|}{\lambda_{k',m'}}\right)\right] \\
&= \sum_{n} E\{z_k^n\} \left[\left(-\frac{1}{\lambda_{k,m}} + \frac{|u_{n,m}|}{(\lambda_{k,m})^2}\right)\right]
\end{aligned}
\tag{14}
$$

Then by setting Eq. (14) to be 0, we get

$$\begin{aligned}
&\sum_n E\{z_k^n\}\left(-\frac{1}{\lambda_{k,m}}+\frac{|u_{n,m}|}{(\lambda_{k,m})^2}\right)=0 \\
&\rightarrow \sum_n E\{z_k^n\}\left(-\lambda_{k,m}+|u_{n,m}|\right)=0 \\
&\rightarrow \sum_n E\{z_k^n\}\lambda_{k,m}=\sum_n E\{z_k^n\}\cdot|u_{n,m}| \\
&\rightarrow \lambda_{k,m}\sum_n E\{z_k^n\}=\sum_n E\{z_k^n\}\cdot|u_{n,m}|
\end{aligned} \tag{15}$$

In summary the maximum of Eq. (11) occurs at the condition in Eq. (16).

$$\lambda_{k,m}=\frac{\sum_n E\{z_k^n\}\cdot|u_{n,m}|}{\sum_n E\{z_k^n\}} \tag{16}$$

2.2.2 Maximization over parameter Π

The gradient method is not appropriate to maximize the log likelihood over parameter Π. The gradient with respect to parameter Π can not simply be set to 0 as shown in Eq.(17) .

$$\frac{\partial E\{\log P(U,Z\,|\,\Lambda,\Pi)\}}{\partial \pi_k}=\frac{\partial \sum_{n,k'} E\{z_{k'}^n\}\left[\log(\pi_{k'})+\sum_{m'}\left(\log(\frac{1}{2\lambda_{k',m'}})-\frac{|u_{n,m'}|}{\lambda_{k',m'}}\right)\right]}{\partial \pi_k}=\sum_n\frac{E\{z_k^n\}}{\pi_k} \tag{17}$$

Instead we need to apply Lagrange Multiplier method. First we can rewrite Eq. (11) as below.

$$\begin{aligned}
E\{\log P(U,Z\,|\,\Lambda,\Pi)\}&=\sum_{n,k'} E\{z_{k'}^n\}\left[\log(\pi_{k'})+\sum_{m'}\left(\log(\frac{1}{2\lambda_{k',m'}})-\frac{|u_{n,m'}|}{\lambda_{k',m'}}\right)\right] \\
&=\sum_{n,k'} E\{z_{k'}^n\}\log(\pi_{k'})+\sum_{n,k'} E\{z_{k'}^n\}\sum_{m'}\left(\log(\frac{1}{2\lambda_{k',m'}})-\frac{|u_{n,m'}|}{\lambda_{k',m'}}\right) \\
&=C+\sum_{n,k'} E\{z_{k'}^n\}\log(\pi_{k'})
\end{aligned} \tag{18}$$

We can define an objective function to maximize and a constraint as

$$f(\Pi)=\sum_{n,k'} E\{z_{k'}^n\}\log(\pi_{k'}) \tag{19}$$

where $\sum_k \pi_k=1$ (from the definition).

The Lagrange function to minimize –f can be written as Eq (20) , where ρ is Lagrange multiplier.

$$L(\Pi,\rho)=-\sum_{n,k'} E\{z_{k'}^n\}\log(\pi_{k'})+\rho(\sum_{k'}\pi_{k'}-1) \tag{20}$$

The minimum of L(Π,ρ) occurs with two conditions,

$$\frac{\partial L(\Pi,\rho)}{\partial \pi_k}=-\sum_n E\{z_k^n\}\frac{1}{\pi_k}+\rho=0 \rightarrow \quad \pi_k=\frac{1}{\rho}\sum_n E\{z_k^n\} \tag{21}$$

$$\frac{\partial L(\Pi,\rho)}{\partial\rho}=\sum_{k'}\pi_{k'}-1=0 \qquad \rightarrow \qquad \sum_{k}\pi_k=1 \tag{22}$$

By substituting Eq. (21) to Eq. (22) we have

$$\sum_k\frac{1}{\rho}\sum_n E\{z_k^n\}=1 \qquad \rightarrow \qquad \sum_k\sum_n E\{z_k^n\}=\rho \tag{23}$$

So we can summarize the maximization condition as below.

$$\pi_k=\frac{\sum_n E\{z_k^n\}}{\sum_k\sum_n E\{z_k^n\}} \tag{24}$$

2.2.3 The EM algorithm

The outline of final EM algorithm is summarized as below.

1. Initialize Z by $\pi_k=\frac{1}{K}$, Λ by $\lambda_{k,m}=E\{|u_m|\}+e$, where e is a small random noise.

2. Calculate the Expectation

$$E\{z_k^n\,|\,U,\Lambda',\Pi'\}=\frac{1}{c_n}\prod_m^M\frac{\pi_k}{2\lambda_{k,m}'}\exp(-\frac{|u_{n,m}|}{\lambda_{k,m}'})$$

3. Maximize the log likelihood given the Expectation

$$\lambda_{k,m}\leftarrow\frac{\sum_n E\{z_k^n\}\cdot|u_{n,m}|}{\sum_n E\{z_k^n\}}\ ,\quad \pi_k\leftarrow\frac{\sum_n E\{z_k^n\}}{\sum_k\sum_n E\{z_k^n\}}$$

4. If (converged) stop, otherwise repeat from step 2.

For the convergence criteria, we can use the expectation of log likelihood, which can be calculated from Eq (11~13). If the value converges and doesn’t change we can stop the iteration. The Expectation of log likelihood has the form of entropy over $E\{z_k^n\}$ and can be calculated easily.

3. Experimental Results

Here we provide examples of image data and show how the learning procedure is performed for the mixture model. We provide visualization of learned variances that reveal the structure of variance correlation, and demonstrate the validity of mixture model by comparing the learned model with the real signal distribution. Finally, we provide simple application examples for image segmentation and Bayesian image denoising.

3.1 Data and Model Learning

As shown in figure 1, our model analyzes image signals in two stages. In the first stage, data vectors are sampled from 16x16 image patches and filtered through ICA forward matrix W to produce source signals. Data samples from image patches are first converted to 256-dimensional vectors and multiplied by ICA matrix W (256 by 160) to produce 160-dimensional source signals. The ICA matrix A and W(=A^{-1}) are learned by Fast ICA algorithm [29]. For training, we randomly sampled 100,000 image patches from a set of natural images gathered from the Internet. Those samples are used again for learning mixture model of at the second stage. Figure 2 shows the visualization of ICA bases.

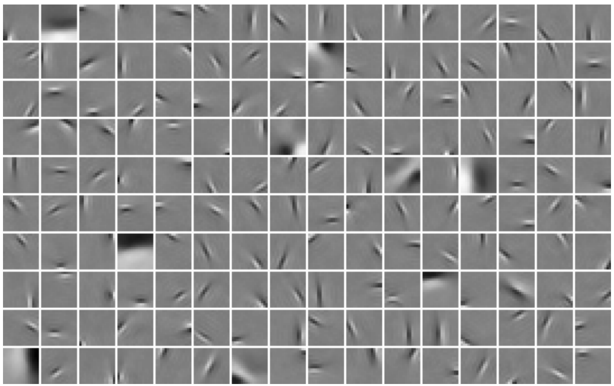
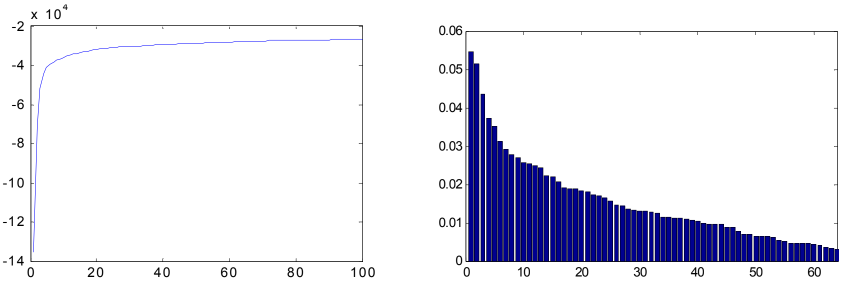


Figure 2. 1st stage ICA basis (Columns of A, 160 dimensions)

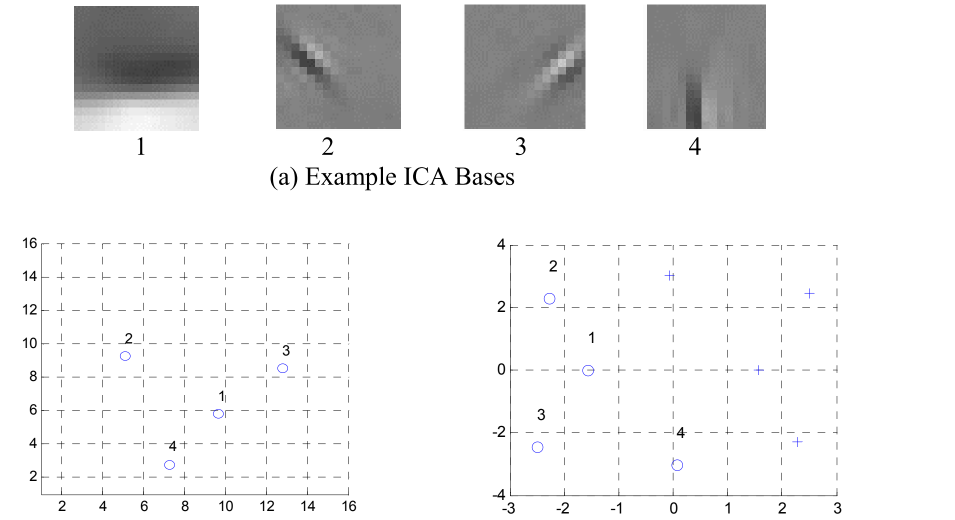
In the second stage, the proposed mixture model is applied to learn the distribution of source signals computed from the first stage ICA. The mixture model has input dimension of 160, and various number of mixtures (16, 64 and 256) are tested.



(a) The log likelihood curve (b) Learned mixture probability (sorted)

Figure 3. EM Learning curve and learned mixture probability. Subfigure (a) shows the plot of log likelihood during 100 EM-iterations. The likelihood value reaches its maximum fast and converges after 40 iterations. Subfigure (b) displays the learned mixture probabilities Π sorted in decreasing order. Those probabilities indicate how often some mixture components appear in the image patch. The number of mixtures is 64 here.

For the visualization of the learned variance, we adapted the visualization method from [22]. Each dimension of ICA source signal corresponds to an image basis (columns of A) and each basis is localized in both image and frequency space with center positions. Figure 4 shows example ICA bases and their corresponding centers in image and frequency space. Since ICA bases have wavelet like shapes, we can compute their center positions in the image and frequency space. For a Laplacian distribution, each component of variance vector corresponds to an image basis. And we can color code the variance values at corresponding center positions in image and frequency space.



(a) Example ICA Bases

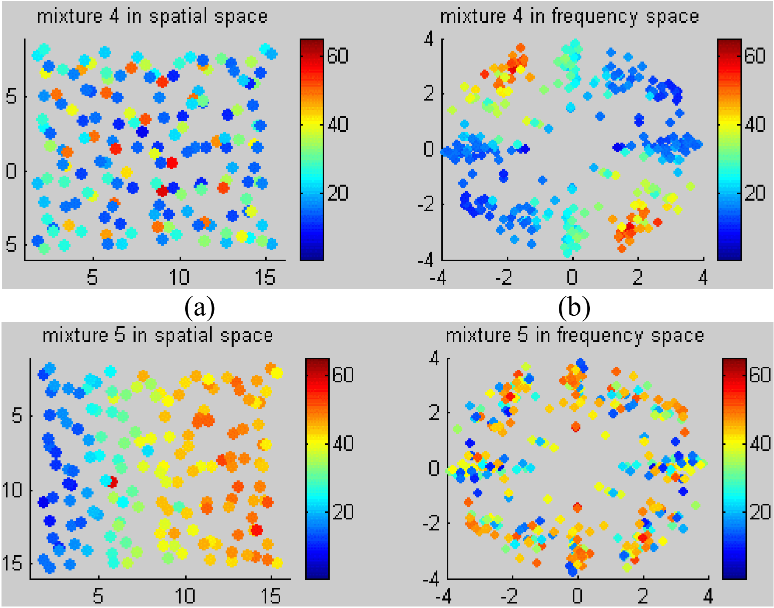
(b) Basis centers in image space

(c) Basis centers in frequency space (Fourier space)

Figure 4. Mapping ICA basis into image space and frequency space. Subfigure (a) shows 4 examples of ICA bases (columns of A matrix). (b) and (c) shows corresponding centers for each basis in image and frequency space (marked by o). Since Fourier transform of an image is origin-symmetric, we added +-marks to denote the mirrored positions for each o-marks.

In figure 4-(c) horizontal axis correspond to vertical frequency and vertical axis corresponds to horizontal frequency space. This axis mapping is more intuitive with respect to edge direction. For example, basis 1 has horizontal edges in image space and its frequency centers are horizontally arranged in the frequency space.

With that information, we can visualize the learned variance vectors for each mixture in image space and frequency space. Figure 5 shows visualization of two mixtures in image and frequency space. Mixture #4 shows strong localization of high variance values in frequency space but there is no localization in image space. We can say this mixture represents textures that mainly consist of oriented edges evenly spread over the image patch. Mixture #5 of figure 5 shows clear localization in image space while no localization in frequency space. This mixture represents image patches whose left side is filled with textures while the right side is relatively clean. This represents boundaries between textured and clean regions.



(a) (b)

(c) (d)

Figure 5. Visualization of learned variances – two examples. Subfigure (a) and (b) visualizes a variance vector for mixture #4 in image and frequency space. (c) and (d) visualizes a variance vector for mixture #5. The high values of variance components are mapped with red color and small values are mapped with blue color.

In the above visualizations, the contrast in frequency or image space is quite 'regular'. i.e. High and low valued variances are smoothly clustered in image or frequency space. Since the neighborhood of ICA source signals are not known to the system, we can say the system learned this regularity from remaining dependencies in the ICA source signal. What is the source of that regularity then ?. A natural image consists of objects with regular textures such as sky, grass, woods and, waters. Those structures can show localized activation of specific orientation, frequency or location in space.

More visualization of learned variance vectors are shown in Figure 6.

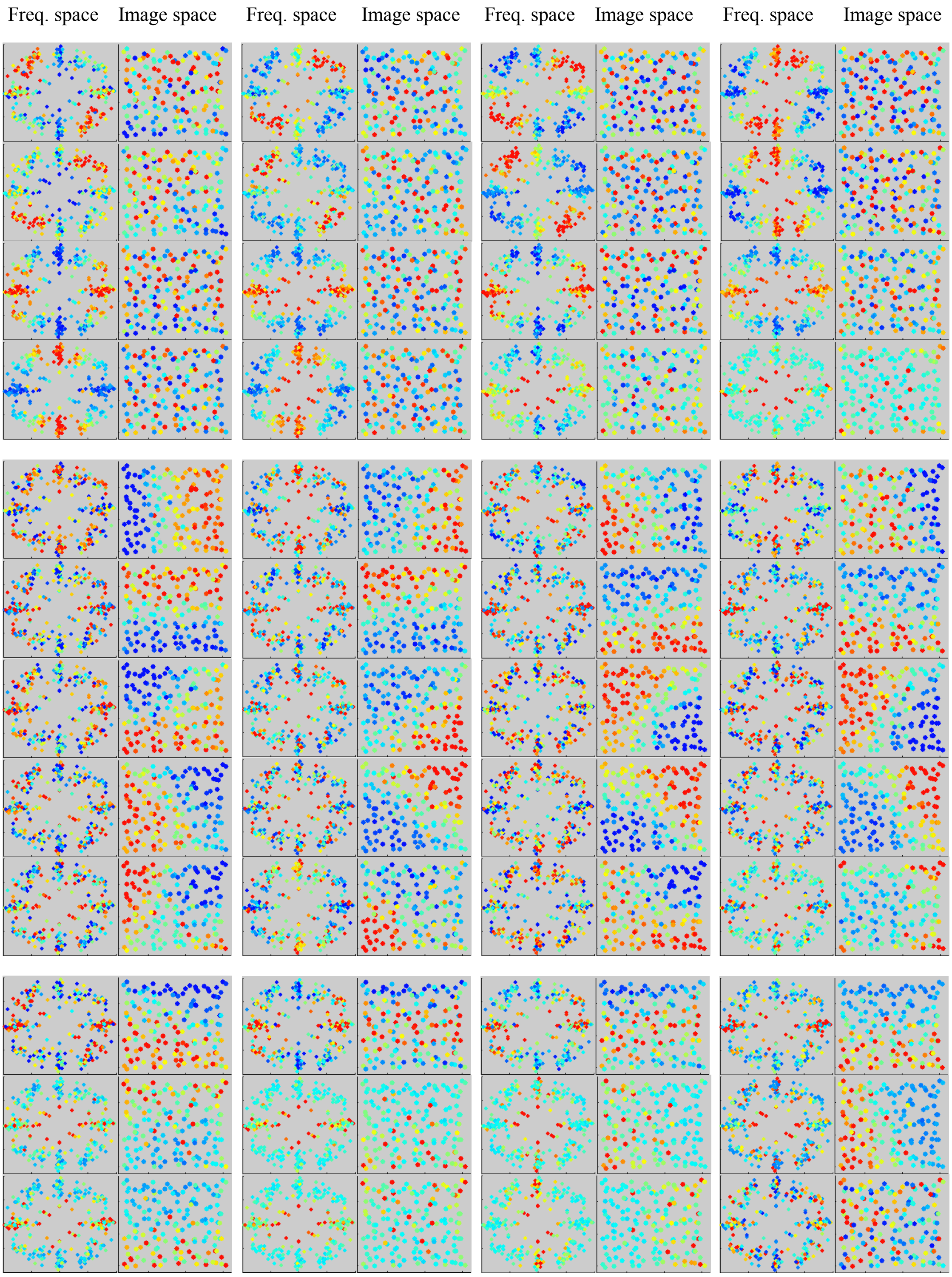


Figure 6. More visualization of learned variance vectors. We grouped mixture variances with frequency-space localization in the first 4 rows. Second 5 rows show variances with image-space localization. Third 3 rows show other types of variances with localization in both image / frequency spaces.

3.2 Comparison between learned and real signal distribution

The proposed model provides a simple parametric form of signal distribution. So we can compare the learned parametric model with real data distribution and explain types non-linearity captured by our mixture model.

Firstly we show how the mixture model captures non-linear structure in 1-D ICA source signal. ICA model assumes a source prior as independent joint distribution of sparse distributions. Laplace distribution is a representative sparse distribution, which can be used as a parametric model of single source signal. As shown in the figure 7-(a) and (c), single Laplacian distribution is a reasonable model of source distribution. But if we look into the detail of figure 7-(c), we notice that the real distributions are more peaked around 0 and probability of large data values are more higher than simple Laplacian distribution. This discrepancy between real data and Laplace distribution can be overcome with the mixture model.

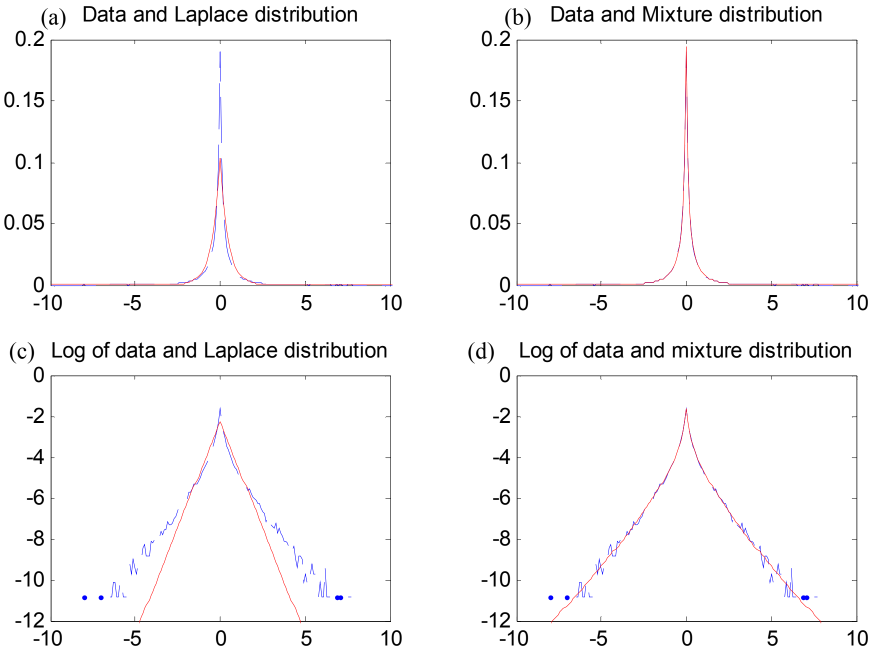


Figure 7. Comparison of single Laplace and mixture distributions with data histogram. (a) Data histogram superimposed with single Laplacian distribution, (b) Data distribution superimposed with Mixture of Laplacian distribution, (c) log plot of plot a, (d) log plot of plot b. (Dashed blue lines denote distribution of real data and solid red lines denote learned parametric distribution. X-axis denotes signal domain. Y-axis denotes the probability or log of probability.)

With the mixture of Laplacian model, we can learn more close approximation of the data distribution as shown in figure 7-(b) and (d). As shown, mixture model can accommodate sharper peak at 0 data value and the increased probability at large data values. This clearly shows how the proposed mixture model can model the ICA source prior better than simple Laplace distribution.

Secondly, we can show how the mixture model can capture the nonlinear dependency between 2-D ICA source signals. Each Laplacian of the learned mixture model captures variance correlation structure. But single Laplacian distribution is just an independent distribution. The key idea of our modeling is that we can mix up independent distributions to get a nonlinearly dependent distribution. Figure 8 shows such modeling power clearly.

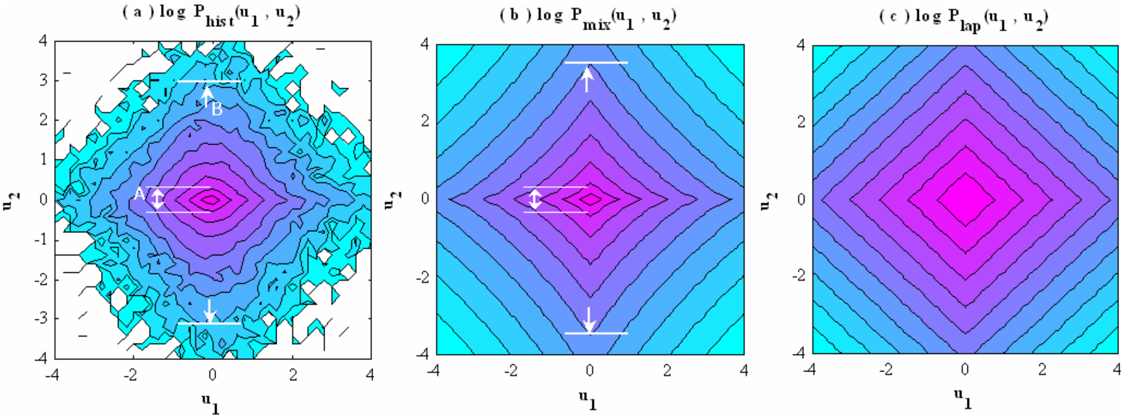


Figure 8: Joint distribution of nonlinearly dependent sources. (a) is a joint histogram of 2 ICA sources, (b) is computed from learned mixture model, and (c) is from learned Laplacian model. In (a), *variance of u_2 is smaller than u_1 at center area (arrow A)*, *but almost equal to u_1 at outside (arrow B)*. So the variance of u_2 is dependent on u_1. This nonlinear dependency is closely approximated by mixture model in (b), but not in (c).

3.3 Application: Unsupervised image segmentation

The idea behind our model is that the image can be modeled as mixture of different variance correlated ‘contexts’. We show how the learned model can be used to classify different context by an unsupervised image segmentation task. We can decide which mixture is mostly responsible for a given image patch by calculating probability of hidden variable Z. The mixture with highest probability is the ones that mostly explain the given image patch. So given model and data, we can compute the expectation of a hidden variable Z from Eq. (12). Then for an image patch, we can select a Laplacian distribution with highest probability, which is the most explaining Laplacian or ‘context’.

Figures 9 to 11 show test images and the segmentation results. We showed only subset of labels for clear visualization. A box at left top corner of the top image denotes the size of image patches used. At the third row we showed color and corresponding mixture variance. Figure 9 and 10 are computed from our model using 64 mixtures. Figure 11 is computed from model with 16 mixtures. The large number of mixtures gives fine grained segmentation of the scene, while the reduced number of mixture gives more abstract segmentation such that, we can see the global and abstract organization of the scene more clearly.

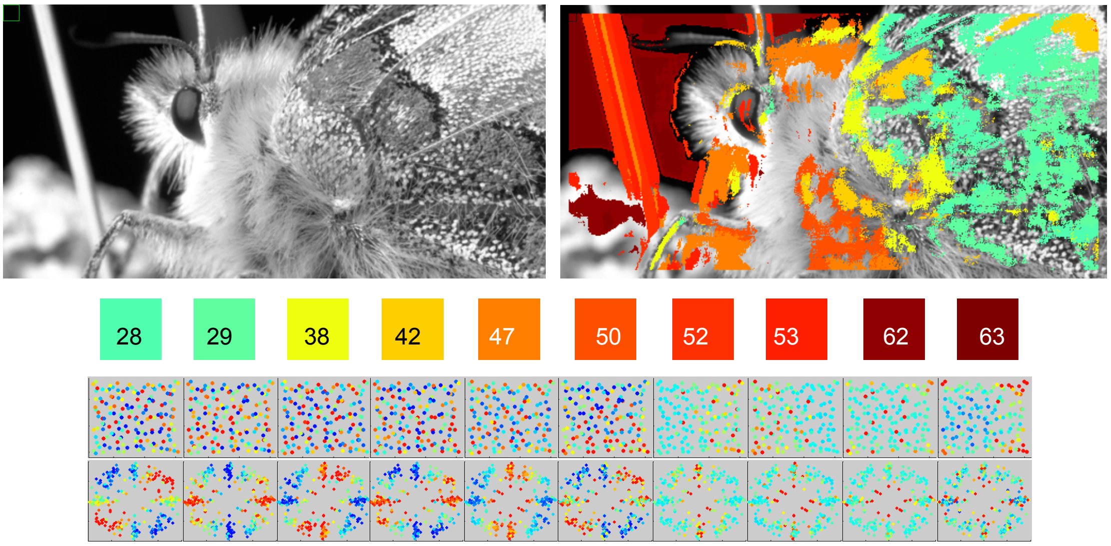


Figure 9. Test image (upper) and color labeled image I (lower), Mixture 28~50 captures textures with high frequency and orientation preference. Mixture 52~62 captures localized low spatial frequency edges (a box on the left top side of original image denotes the patch window size).

Figure 10 clearly shows correlation between image structures and variance correlated.

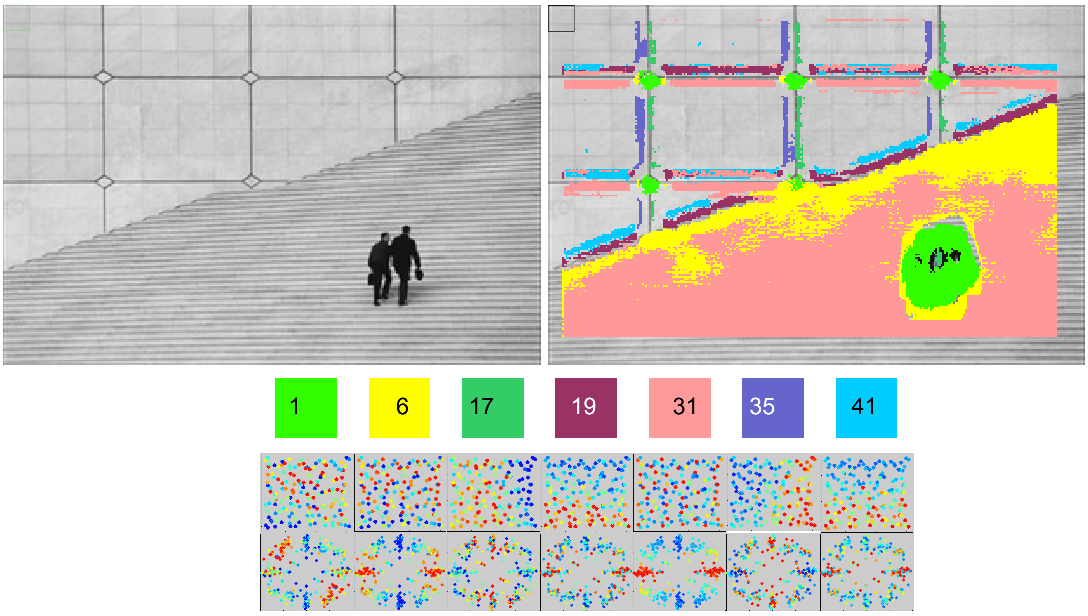


Figure 10. Test image and color labeled image II, Mixture 1 captures patches with high frequency multi-oriented edges. Mixture 31 responses to (repeated) horizontal edge structures. Mixture 6 is similar to mixture 31 but respond to more high frequency horizontal edges. Mixture label 17 and 35 captures vertical edges.

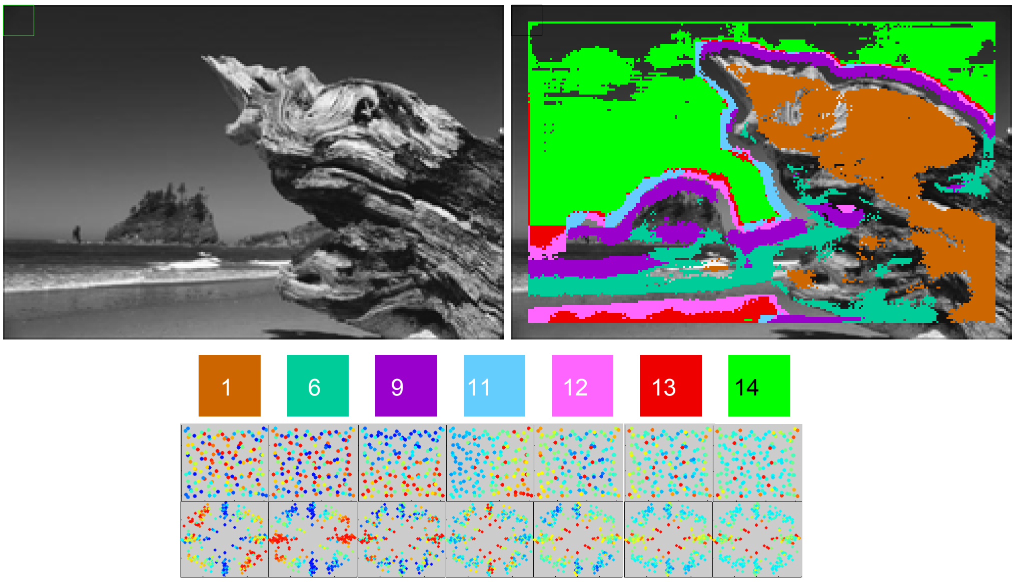


Figure 11. Test image and color labeled image III. Mixture 1 captures high frequency texture structures caused by wood. Mixture 14 captures sky and mixture 9 captures the boundary between clear sky and textured land scene.

As shown in results, the mixture model can classify image contexts, which are highly correlated with the regular structures or objects in the scene. Also the labeling shows high degree of regularity across image space and the regularity can be further analyzed to extract more high-level features.

Also our mixture model discovers new regularity over image space. In figures 9~11, the labeling results are much continuous or smooth. This is due to the smoothness of natural contexts since object surfaces tend to be clustered in image space. For example, stairs in figure 8 and wood texture in figure 9 tend to occur continuously across space. This implies that our mixture model learned invariant features. Also such 'invariance' of representation is an important feature of biological information processing.

3.4 Application: Image Denoising

The proposed mixture model provides a better parametric model for ICA source prior and hence an improved model of the image patches. We can take advantage of this in Bayesian MAP (maximum a posteriori) estimation for image denoising.

ICA has been used for image denoising in Bayesian framework [1][3]. In that framework, we assume an additive noise n is added to original image as shown in Eq. 25. Then, we can obtain the MAP estimation of ICA source signal u using Eq. (26) and reconstruct a cleaned image by computing $\hat{X} = A\hat{u}$.

$$X = Au + n \qquad , \quad n \sim N(0, \sigma_n^2) \tag{25}$$

$$\hat{u} = \underset{u}{argmax} \ \log P(u \mid X, A) = \underset{u}{argmax} \left[\log P(X \mid u, A) + \log P(u) \right] \tag{26}$$

In Eq. (26), $P(X \mid u, A)$ is a Gaussian distribution since we assumed Gaussian noise. But the source prior $P(u)$ can be varied depending on source model [1][3][30][31][32]. For ICA with Laplacian prior, we can rewrite Eq. (26) as Eq. (27). Let's call this method as ICA MAP with Laplacian prior [3].

$$\hat{u} = \underset{u}{argmax}\left(-\frac{|X - Au|^2}{2\sigma_n{}^2} - \sum_m \frac{|u_m|}{\lambda_m}\right) \tag{27}$$

But we can use more refined prior based on our mixture model. For efficient computation, we can approximate mixture prior by single most explaining Laplacian distribution k as shown in Eq. (28). The most probable mixture k can be computed from image segmentation task. We call this new estimation method as ICA MAP with Mixture prior.

$$\hat{u} = \underset{u}{argmax}\left(-\frac{|X - Au|^2}{2\sigma_n{}^2} - \sum_m \frac{|u_m|}{\lambda_{k,m}}\right) \tag{28}$$

There is no known analytical solution to maximization problem of Eq. (27~ 28). In general, we have to use gradient descent method, which can be slow and suboptimal. Instead, we can use orthogonal ICA transform (W=A^T) where an efficient analytic solution is available. Hyvarinen [3] suggested orthogonalizing ICA transform W through $W_o = W(W^TW)^{-1/2}$. This orthogonalized bases have similar shape as the bases of original W, and it allows deterministic solution to Eq.(27~28). Since W_o is also orthonormal, we have $A_o \equiv W_o{}^{-1} = W_o{}^T$ and $A_o{}^T = W_o$. This simplifies the multidimensional maximization problem into independent 1-D maximization problems as shown in Eq. (29). The solution to this 1-D maximization problem is reported in [3].

$$\begin{aligned}
&-\frac{|X - A_o u|^2}{2\sigma_n{}^2} - \sum_i \frac{|u_i|}{\lambda_{k,i}} = -\tfrac{1}{2\sigma_n{}^2}(X - A_o u)^T(X - A_o u) - \sum_i \frac{|u_i|}{\lambda_{k,i}} \\
&= -\tfrac{1}{2\sigma_n{}^2}(X - A_o u)^T(A_o A_o{}^T)(X - A_o u) - \sum_i \frac{|u_i|}{\lambda_{k,i}} \\
&= -\tfrac{1}{2\sigma_n{}^2}(A_o{}^T X - A_o{}^T A_o u)^T(A_o{}^T X - A_o{}^T A_o u) - \sum_i \frac{|u_i|}{\lambda_{k,i}} \\
&= -\tfrac{1}{2\sigma_n{}^2}(A_o{}^T X - u)^T(A_o{}^T X - u) - \sum_i \frac{|u_i|}{\lambda_{k,i}} \\
&= -\tfrac{1}{2\sigma_n{}^2}|W_o X - u|^2 - \sum_i \frac{|u_i|}{\lambda_{k,i}} = -\tfrac{1}{2\sigma_n{}^2}|u' - u|^2 - \sum_i \frac{|u_i|}{\lambda_{k,i}} = -\sum_i\left(\tfrac{1}{2\sigma_n{}^2}(u'_i - u_i)^2 + \frac{|u_i|}{\lambda_{k,i}}\right)
\end{aligned} \tag{29}$$

In addition to two ICA based methods, we also evaluated 3 other methods including BayesCore [31], BayesJoint [32], and Wiener filter [33]. BayesCore is similar to ICA MAP method except that it uses wavelet transformation and Bayes marginal estimator instead of MAP estimator [31]. BayesJoint is also a Bayes marginal estimator with wavelet transformation and Gaussian source prior where the variances of sources are linearly correlated with each other [32]. Although the BayesJoint method is one of the first to consider variance correlation for denoising, its modeling power is much limited than our model. It assumes only linear correlation of variances while our model can learn nonlinear correlation. We summarize the differences between those denoising methods in Table 1.

Denoising Method	Generative Basis	Source Prior	Estimation Method
Wiener Filter	Fourier	Gaussian	Bayes marginal
BayesCore	Wavelet	Generalized Laplacian	Bayes marginal
BayesJoint	Wavelet	Correlated Gaussian*	Bayes marginal
ICA MAP (Laplacian prior)	ICA	Laplacian	Bayes MAP
ICA MAP (Mixture prior)	ICA	Mixture of Laplacian*	Bayes MAP

Table 1. Comparison of different denoising methods tested in experiments. (ICA MAP with mixture prior is the proposed method. Priors that model variance correlation are marked with *.)

General settings for the experiments are like this. For all denoising methods, we assume that the noise variance is known. Then we can compute the variance of original image by subtracting the noise variance from variance of noisy image. Using the orthogonal ICA matrix, we trained a new mixture model with 256 Laplacian distributions. Then denoising experiments are performed with test images different from model training images. All the test images are pre-normalized to have unit variance. For both ICA based denoising methods, a 16x16 estimation window is shifted over the image by every 2 pixels. Then the overlapping reconstructions are averaged to produce final results. For both wavelet-based methods, we use QMF(quadratic mirror filter) type wavelet decomposition.

For measuring denoising performance, we used signal to noise ratio (SNR) and structural similarity measure (SSIM) [34]. SSIM is an image similarity measure, which is closely matched with human perceptual similarity. It measures similarity of two images as a composition of luminance, contrast and structural similarity and the measure of similarity varies from 0 to 1.

Figure 12 shows an example image ‘Beach’ and its denoising results. Subfigures 12-(d~h) show results from 5 different denoising methods. Subfigure 12-(c) is a segmentation result that is used for denoising by ICA MAP with mixture prior.

Figure 13 summarizes the denoising performance of 5 algorithms on the image ‘Beach’. Denoising by ICA MAP with mixture prior outperforms other methods. Especially our denoising method is much better in terms of perceptual quality. Denoised image in figure 12-(h) looks clearer and sharper than other denoising results. This is because our algorithm estimates and utilizes variance correlation structure for each patch. This refined prior model helps to preserve edges and structural information while most Gaussian noise components are removed. Other methods such as Wiener and BayesJoint consider signal and noise to be Gaussian, and separating the Gaussian noise component from Gaussian signal is more ambiguous. Figure 14 and 15 shows results of another test on ‘Lena’ image. The results are similar to that of the ‘Beach’ image.

In summary, denoising based on our model outperformed other methods. ICA MAP with simple Laplace prior and BayesCore methods assumes sparse independent prior and performed similarly. BayesJoint method performed better than BayesCore in terms of perceptual quality, but still worse than our method. All ICA and Wavelet based methods performed better than Wiener filter, which assumes Gaussian source prior.

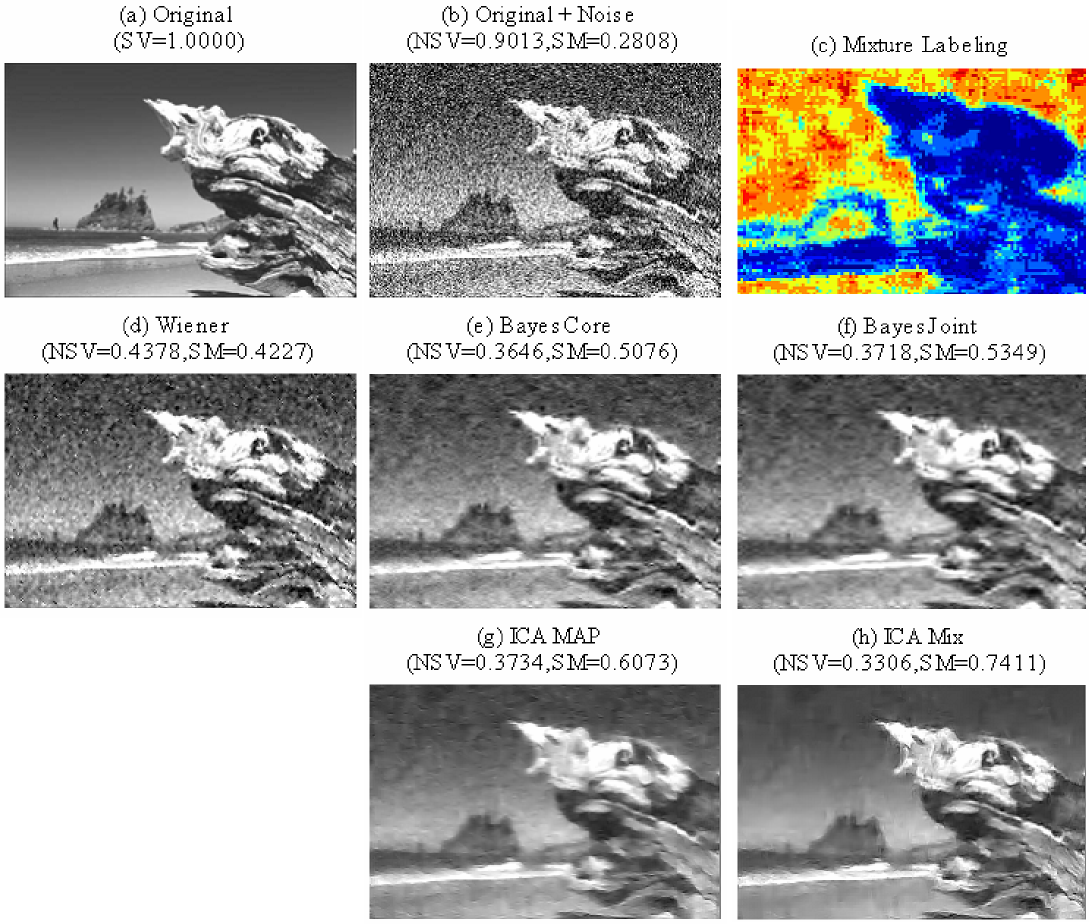


Figure 12: Example denoising results of 'beach' image (image var. = 1.0, input noise standard variance = 0.9, NSV denotes noise standard variance, and SM denotes structural similarity index.)

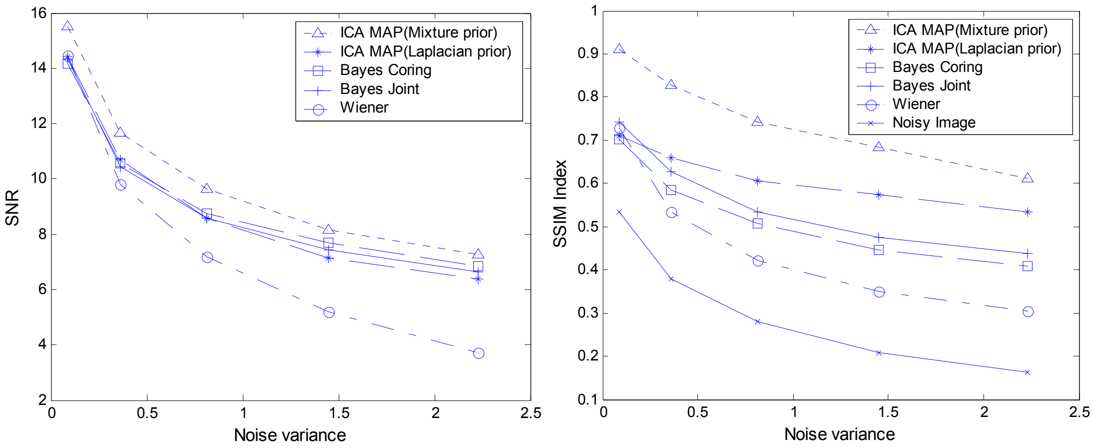
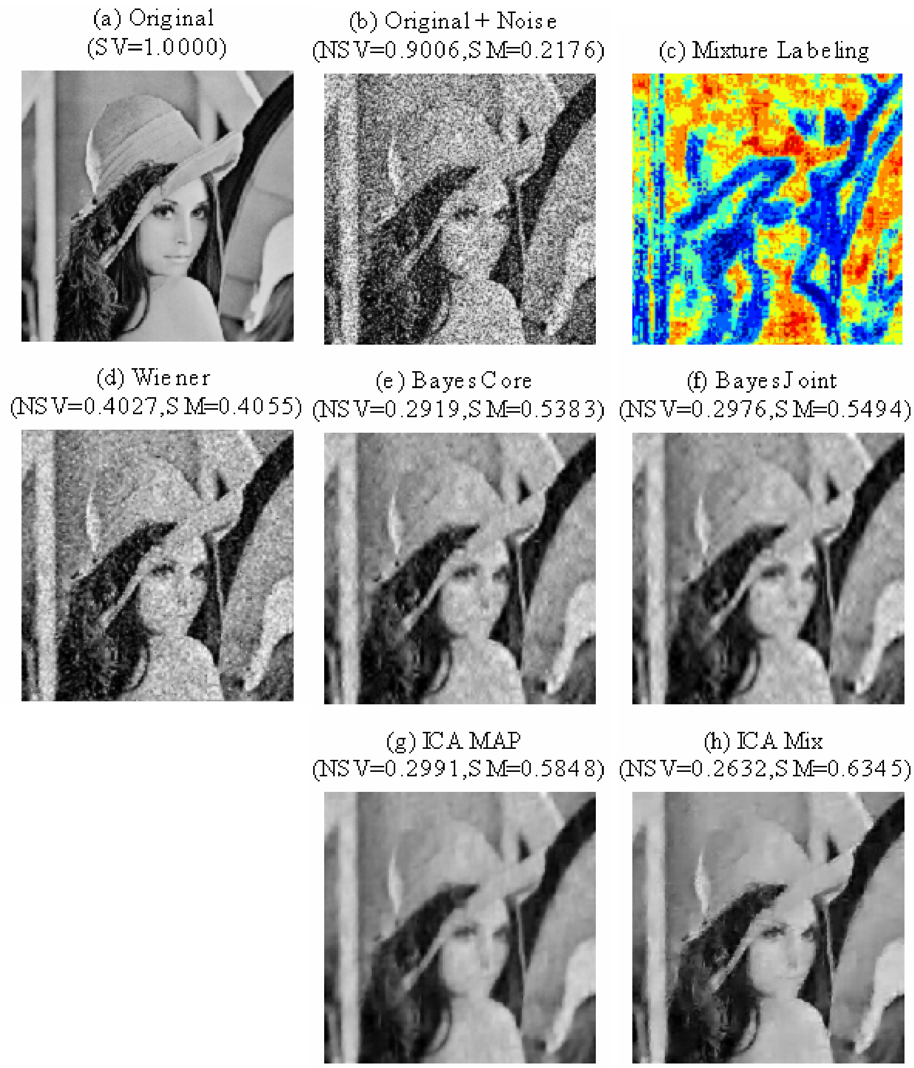


Figure 13: SNR and SSIM for 'beach' image over different noise variances (image var._= 1.0, SSIM index is 1 if the denoised image is same as the original image, and 0 ≤ SSIM index ≤ 1)



(a) Original (SV=1.0000)

(b) Original + Noise (NSV=0.9006,SM=0.2176)

(c) Mixture Labeling

(d) Wiener (NSV=0.4027,SM=0.4055)

(e) BayesCore (NSV=0.2919,SM=0.5383)

(f) BayesJoint (NSV=0.2976,SM=0.5494)

(g) ICA MAP (NSV=0.2991,SM=0.5848)

(h) ICA Mix (NSV=0.2632,SM=0.6345)

Figure 14: Example denoising results of ‘Lena’ image (image var. = 1.0, input noise standard variance = 0.9, NSV denotes noise standard variance, and SM denotes structural similarity index.)

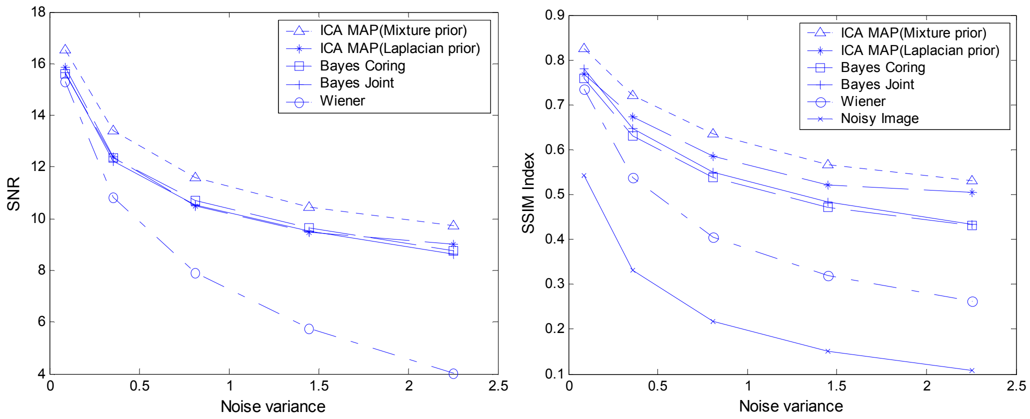


Figure 15: SNR and SSIM for Lena Image over different noise variances (image variance = 1.0)

4. Discussion

We proposed a mixture model to learn nonlinear dependencies of ICA source signals for natural images. The proposed mixture of Laplacian distribution model is a generalization of the conventional independent source priors and can model variance dependency given natural image signals. Experimental results show that the proposed model can learn variance correlated signal group as different mixtures and learn high-level structures, which are highly correlated with underlying physical properties. Furthermore our model provides an analytic prior of nearly independent and variance-correlated signals, which is not viable in previous models [19][20][21][22][5][6].

The learned variance vectors of the mixture model show structured localization in image and frequency space, which is similar to the result in [22]. Since the model is given no information about the spatial location or frequency of the source signals, we can say that the dependency captured by the mixture model reveal abundant regularity in natural images. That regularity is the cause of remaining nonlinear dependencies in the ICA source signals. As shown in the image segmentation experiments, such regularities correspond to specific surface types (textures) or boundaries between surfaces.

The variance correlation is believed to reflect higher order structures in natural signal. Statistical modeling that accommodates such high order structures can contribute to the understanding of natural image statistics and the neural information processing in the brain. Also those high-level representations learned by our proposed model can be used as features for image segmentation, pattern recognition, as well as image coding. In this paper we demonstrated initial applications to image segmentation and denoising.

We demonstrate that the learned mixture model can be used to discover hidden contexts that generated regularity in the image. From image segmentation experiments, we verified that labeling of the image patches based on learned mixture model is highly correlated with the object surface types shown in the image. Also the results show that such contexts are clustered in image space. This type of regularity is discovered by the model in unsupervised manner and can be exploited further for discovering even higher-level concepts.

Finally, we showed applications of our model for image denoising in Bayesian framework. We illustrated its performance and compared it with 4 other conventional methods. Our results suggest that the proposed model outperforms other methods. It is due to the estimation of the correlated variance structure, which provides an improved prior that has not been considered in other methods [3][31][32][33].

There are also some limitations in our current model. Our model is not fully adaptive. i.e. we fixed the first stage representation (linear generative matrix, A) during the learning of the second stage (prior of source signal). We assumed that the source signals are nonlinearly correlated at the second stage, but the first stage representation was learned based on independence assumption. In the second stage, we assumed the source codes are independent given a specific Laplacian distribution and variance. There is a small mismatch between assumptions of second stage and first stage. But as shown in the denoising experiments, our model works because the discrepancy in the assumptions is small. i.e. ICA is already a good approximation of nearly independent signals. But it might be a better statistical model if we can adapt both the first and second stage.

For further investigation, there can be several ways to extend our model. First, we can exploit the regularity of the image segmentation result to learn higher-level structures by building additional hierarchies on the current model. As shown in the image segmentation experiment, the mixture labeling shows much continuity over image space. This reveals another type of dependency and correlation in the image signal. And this dependency can be learned though a third stage model which can lead to even higher representation such as ‘objects’ and ‘parts’.

Second, we can extend our model to fully generative model by adapting both the first stage (linear generative matrix) and second stage (source mixture prior). Even though the changes can be small, we can still have a better model of image statistics.

Thirdly, there can be more applications of our model. The application to image coding seems promising since our model provides a better analytic prior. Application to image segmentation should be exploited further. Also our model can be used for texture analysis and synthesis as the texture types are described well by the different mixtures.

Acknowledgments

This work was in part supported by the Post-doctoral Fellowship Program of Korea Science & Engineering Foundation (KOSEF).

References

[1] G. E. Hinton, & T. J. Sejnowski, *Unsupervised Learning: Foundations of Neural Computation* (edited), MIT Press, Cambridge, Massachusetts, 1999.

[2] M.S. Lewicki, & B.A. Olshausen (1999), Probabilistic Framework for the Adaptation and Comparison of Image Codes, Journal of the Optical Society of America, 16(7): 1587-1601.

[3] A. Hyvarinen, Sparse Code Shrinkage: Denoising of Nongaussian Data by Maximum Likelihood Estimation,Neural Computation, 11(7):1739-1768, 1999.

[4] A. Hyvarinen, P.O. Hoyer, and E. Oja, Image Denoising by Sparse Code Shrinkage, In S. Haykin and B. Kosko (eds), Intelligent Signal Processing, IEEE Press, 2001.

[5] T. W. Lee, M.S. Lewicki, and T.J. Sejnowski., ICA Mixture Models for unsupervised Classification of non-gaussian classes and automatic context switching in blind separation. PAMI, 22(10), October 2000.

[6] T.W., Lee and M.S. Lewicki, Unsupervised Image Classification, Segmentation, and Enhancement Using ICA Mixture Models , IEEE Transactions On Image Processing, Vol. 11, No. 3, March 2002.

[7] H. Barlow, *Sensory Communication*. Cambridge, MA: MIT Press, 1961, pp. 217–234.

[8] D.J. Field, What is the goal of sensory coding?, *Neural Comput.*, vol. 6, pp. 559–601, 1994.

[9] D.J. Field, Relations between the statistics of natural images and the response properties of cortical cells. J. Opt. Soc. Am. A 4, 2379-2394, 1987.

[10] B.A. Olshausen and D.J. Field (1996) Emergence of simple-cell receptive-field properties by learning a sparse code for natural images, Nature, 381, 607–9.

[11] B.A Olshausen and D.J. Field (1996). Natural Image Statistics and Efficient Coding, *Network, 7*: 333-339.

[12] A. J. Bell and T. J. Sejnowski, The 'Independent Components' of Natural Scenes are Edge Filters, *Vision Research*, 37(23):3327–3338, 1997.

[13] J. H. van Hateren and A_ van der Schaaf, Independent component filters of natural images compared with simple cells in primary visual cortex, Proc. Royal Soc. Lond. B. 265:359-336, 1998.

[14] A.J. Bell and T.J. Sejnowski (1995), An information-maximization approach to blind separation and blind deconvolution, Neural Computation 7:1129-1159.

[15] T.W. Lee (1998), Independent Component Analysis: Theory and Applications, Kluwer Academic Publishers, Boston, ISBN: 0 7923 8261 7, September 1998.

[16] A. Hyvärinen, J. Karhunen and E. Oja (2002), Independent Component Analysis, John Wiley and Sons.

[17] J.F. Cardoso (1997) Infomax and maximum likelihood for blind source separation *IEEE Signal Process. Lett.* **4** 109–11.

[18] E.W. Weisstein, Laplace Distribution. From MathWorld--A Wolfram Web Resource. http://mathworld.wolfram.com/LaplaceDistribution.html.

[19] A. Hyvarinen, P. O. Hoyer., Emergence of phase and shift invariant features by decomposition of natural images into independent feature subspaces, Neural Computation, 12 (7) 1705-1720, 2000.

[20] Aapo Hyvarinen, Patrik O. Hoyer , Topographic Independent component analysis as a model of V1 Receptive Fields, *Neurocomputing*, Vol. 38-40, 1307-1315, 2001.

[21] Aapo Hyvarinen, Beyond Independent Component Analysis, ICANN 99 , 9th, Vol. 2, pp 809-814.

[22] M. S. Lewicki and Y. Karklin (2003), Learning higher-order structures in natural images, Network: Comput. Neural Syst. 14 (August 2003) 483-499.

[23] Odelia Schwartz and Eero P. Simoncelli, Natural signal statistics and sensory gain control, Nature Neuroscience, 2001, August.

[24] J Wainwright, O Schwartz, and E P Simoncelli, Natural image statistics and divisive normalization: Modeling nonlinearity and adaptation in cortical neurons. In Probabilistic Models of the Brain: Perception and Neural Function, eds. MIT Press. Spring, 2002.

[25] M. Welling and G. E. Hinton, S. Osindero, Learning Sparse Topographic Representations with Products of Student-t Distributions, NIPS, 2002.

[26] J. Portilla, V. Strela, M. J. Wainwright and E. P Simoncelli, Image Denoising using Scale Mixtures of Gaussians in the Wavelet Domain, IEEE Trans. On Image Processing, Vol.12, No. 11, 1338-1351, 2003.

[27] A.P. Dempster, N.M. Laird and D.B. Rubin, (1977) Maximum likelihood from incomplete data via EM algorithm. Journal of the Royal Statistical Society 39:1-38.

[28] T. M. Mitchell, *Machine Learning*, Carnegie Mellon Univ., McGraw-Hill. 1997.

[29] A. Hyvarinen and E. Oja, A fast fixed-point algorithm for independent component analysis, *Neural Comput.*, vol. 9, pp. 1483–1492, 1997.

[30] D. L. Donoho and I. M. Johnstone, Threshold selection for wavelet shrinkage of noisy data, Proceedings of the 16th Annual International Conference of the IEEE, vol. 1, 1994, pp. A24-A25.

[31] E. P. Simoncelli and E. H. Anderson, Noise removal via Bayesian wavelet coring, Proceedings of the 3rd IEEE International Conference on Image Processing, vol. 1, 1996, pp. 379-382.

[32] E. P. Simoncelli, Bayesian Denoising of Visual Images in the Wavelet Domain, Bayesian Inference in Wavelet Based Models, eds. P Muller and B Vidakovic, Chapter 18, Springer-Verlag, 1999, pp 291-308.

[33] A. K. Jain, *Fundamentals of Digital Image Processing*, Prentice-Hall, Englewood Cliffs, New Jersey, 1989.

[34] Z. Wang, A. C. Bovik, H. R. Sheikh and E. P. Simoncelli, Image quality assessment: From error visibility to structural similarity, IEEE Transactions on Image Processing, vol. 13, no. 4, Apr. 2004.