Int'l Journal of Computer Vision 27(2) 1-20, March/April, 1998.

Filters, Random fields And Maximum Entropy (FRAME)

—Towards a Unified Theory for Texture Modeling

Song Chun Zhu [1] Yingnian Wu [2] David Mumford[1]

Abstract

This article presents a statistical theory for texture modeling. This theory combines filtering theory and Markov random field modeling through the maximum entropy principle, and interprets and clarifies many previous concepts and methods for texture analysis and synthesis from a unified point of view. Our theory characterizes the ensemble of images $\mathbf{I}$ with the same texture appearance by a probability distribution $f(\mathbf{I})$ on a random field, and the objective of texture modeling is to make inference about $f(\mathbf{I})$, given a set of observed texture examples. In our theory, texture modeling consists of two steps. (1) A set of filters is selected from a general filter bank to capture features of the texture, these filters are applied to observed texture images, and the histograms of the filtered images are extracted. These histograms are estimates of the marginal distributions of $f(\mathbf{I})$. This step is called feature extraction. (2) The maximum entropy principle is employed to derive a distribution $p(\mathbf{I})$, which is restricted to have the same marginal distributions as those in (1). This $p(\mathbf{I})$ is considered as an estimate of $f(\mathbf{I})$. This step is called feature fusion. A stepwise algorithm is proposed to choose filters from a general filter bank. The resulting model, called FRAME (Filters, Random fields And Maximum Entropy), is a Markov random field (MRF) model, but with a much enriched vocabulary and hence much stronger descriptive ability than the previous MRF models used for texture modeling. Gibbs sampler is adopted to synthesize texture images by drawing typical samples from $p(\mathbf{I})$, thus the model is verified by seeing whether the synthesized texture images have similar visual appearances to the texture images being modeled. Experiments on a variety of 1D and 2D textures are described to illustrate our theory and to show the performance of our algorithms. These experiments demonstrate that many textures which are previously considered as from different categories can be modeled and synthesized in a common framework.

[1] Division of Applied Math, Brown University, Providence, RI 02912

[2] Department of Statistics, University of Michigan, Ann Arbor, MI 48109

1 Introduction

Texture is an important characteristic of the appearance of objects in natural scenes, and is a powerful cue in visual perception. It plays an important role in computer vision, graphics, and image encoding. Understanding texture is an essential part of understanding human vision.

Texture analysis and synthesis has been an active research area during the past three decades, and a large number of methods have been proposed, with different objectives or assumptions about the underlying texture formation processes. For example, in computer graphics, reaction-diffusion equations (Witkin and Kass 1991) have been adopted to simulate some chemical processes that may generate textures on skin of animals. In computer vision and psychology, however, instead of modeling specific texture formation process, the goal is to search for a general model which should be able to describe a wide variety of textures in a common framework, and which should also be consistent with the psychophysical and physiological understanding of human texture perception.

The first general texture model was proposed by Julesz in the 1960's. Julesz suggested that texture perception might be explained by extracting the so-called 'k-th order' statistics, i.e., the co-occurrence statistics for intensities at k-tuples of pixels (Julesz 1962). Indeed, early works on texture modeling were mainly driven by this conjecture (Haralick 1979). A key drawback for this model is that the amount of data contained in the k-th order statistics is gigantic and thus very hard to handle when $k > 2$. On the other hand, psychophysical experiments show that the human visual system does extract at least some statistics of order higher than two (Diaconis and Freeman 1981).

More recent work on texture mainly focus on the following two well-established areas.

One is filtering theory, which was inspired by the multi-channel filtering mechanism discovered and generally accepted in neurophysiology (Silverman et al. 1989). This mechanism suggests that visual system decomposes the retinal image into a set of sub-bands, which are computed by convolving the image with a bank of linear filters followed by some nonlinear procedures. The filtering theory developed along this direction includes the Gabor filters (Gabor 1946, Daugman 1985) and wavelet pyramids (Mallat 1989, Simoncelli etc. 1992, Coifman and Wickerhauser 1992, Donoho and Johnstone 1994). The filtering methods show excellent performance in classification and segmentation (Jain and Farrokhsia 1991).

The second area is statistical modeling, which characterizes texture images as arising from probability distributions on random fields. These include time series models (McCormick and Jayaramamurthy 1974), Markov chain models (Qian and Terrington 1991), and Markov random field (MRF) models (Cross and Jain 1983, Mao and Jain 1992, Yuan and Rao 1993). These modeling approaches involve only a small number of parameters, thus provide concise representation for textures. More importantly, they pose texture analysis as a well-defined statistical inference problem. The statistical theories enable us not only to make inference about the parameters of the underlying probability models based on observed texture images, but also to synthesize texture images by sampling from these probability models. Therefore, it provides a rigorous way to test the model by checking whether the synthesized images have similar visual appearances to the textures being modeled (Cross and Jain 1983). But usually these models are of very limited forms, hence suffer from the lack of expressive power.

This paper proposes a modeling methodology which is built on and directly combines the above two important themes for texture modeling. Our theory characterizes the ensemble of images $\mathbf{I}$ with the same texture appearances by a probability distribution $f(\mathbf{I})$ on a random field. Then given a set of observed texture examples, our goal is to infer $f(\mathbf{I})$. The derivation of our model consists of two steps.

(I) A set of filters is selected from a general filter bank to capture features of the texture. The filters are designed to capture whatever features might be thought to be characteristic of the given texture. They can be of any size, linear or nonlinear. These filters are applied to the observed texture images, and histograms of the filtered images are extracted. These histograms estimate the marginal distributions of $f(\mathbf{I})$. This step is called *feature extraction*.

(II) Then a maximum entropy distribution $p(\mathbf{I})$ is constructed, which is restricted to match the marginal distributions of $f(\mathbf{I})$ estimated in step (I). This step is called *feature* *fusion.*

A stepwise algorithm is proposed to select filters from a general filter bank, and at each step k it chooses a filter $F^{(k)}$ so that the marginal distributions of $f(\mathbf{I})$ and $p(\mathbf{I})$ with respect to $F^{(k)}$ have the biggest distance in terms of L_1 norm. The resulting model, called **FRAME** (*Filters, Random fields And Maximum Entropy*), is a Markov random field (MRF) model [1], but with a much more enriched vocabulary and hence much stronger descriptive power compared with previous MRF models. The Gibbs sampler is adopted to synthesize texture images by drawing samples from $p(\mathbf{I})$, thus the model is tested by synthesizing textures in both 1D and 2D experiments.

Our theory is motivated by two aspects. Firstly, a theorem proven in section (3.2) shows that a distribution $f(\mathbf{I})$ is uniquely determined by its marginals. Therefore if a model $p(\mathbf{I})$ matches all the marginals of $f(\mathbf{I})$, then $p(\mathbf{I}) = f(\mathbf{I})$. Secondly, recent psychophysical research on human texture perception suggests that two 'homogeneous' textures are often difficult to discriminate when they have similar marginal distributions from a bank of filters (Bergen and Adelson 1991, Chubb and Landy 1991). Our method is inspired by and bears some similarities to Heeger and Bergen's (1995) recent work on texture synthesis, where many natural looking texture images were synthesized by matching the histograms of filter responses organized in the form of a pyramid.

This paper is arranged as follows. First we set the scene by discussing filtering methods and Markov random field models in section (2), where both the advantages and disadvantages of these approaches are addressed. Then in section (3), we derive our FRAME model and propose a feature matching algorithm for probability inference and stochastic simulation. Section (4) is dedicated to the design and selection of filters. The texture modeling experiments are divided into three parts. Firstly section (5) illustrates important concepts of the FRAME model by designing three experiments for one dimensional texture synthesis. Secondly a variety of 2D textures are studied in section (6). Then section (7) discusses a special diffusion strategy for modeling some typical texton images. Finally section (8) concludes with a discussion and the future work.

[1]Among statisticians, MRF usually refers to those models where the Markov neighborhood is very small, e.g. 2 or 3 pixels away. Here we use it for any size of neighborhood.

2 Filtering and MRF Modeling

2.1 Filtering theory

In the various stages along the visual pathway, from retina, to V1, to extra-striate cortex, cells with increasing sophistication and abstraction have been discovered: center-surround isotropic retinal ganglion cells, frequency and orientation selective simple cells, and complex cells that perform nonlinear operations. Motivated by such physiological discoveries, the filtering theory proposes that the visual system decomposes a retinal image into a set of sub-band images by convolving it with a bank of frequency and orientation selective linear filters. This linear filtering process is then followed by some nonlinear operations. In the design of various filters, Gaussian function plays an important role due to its nice low-pass frequency property. To fix notation, we define an elongated two-dimensional Gaussian function as:

$$G(x, y \mid x_0, y_0, \sigma_x, \sigma_y) = \frac{1}{2\pi\sigma_x\sigma_y} e^{-((x-x_0)^2/2\sigma_x^2 + (y-y_0)^2/2\sigma_y^2)}$$

with location parameters (x_0, y_0) and scale parameters (σ_x, σ_y).

A simple model for the radially symmetric center-surround ganglion cells is the Laplacian of Gaussian with $\sigma_x = \sigma_y = \sigma$:

$$F(x, y \mid x_0, y_0, \sigma) = (\frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2})G(x, y \mid x_0, y_0, \sigma, \sigma). \tag{1}$$

Similarly, a model for the simple cells is the Gabor filter (Daugman 1985), which is a pair of cosine and sine waves with frequency ω and amplitude modulated by the Gaussian function:

$$F_\omega(x, y) = G(x, y \mid 0, 0; \sigma_x, \sigma_y)e^{-i\omega x}. \tag{2}$$

By carefully choosing the frequency ω and rotating the filter in the x-y coordinate system, we obtain a bank of filters which cover the entire frequency domain. Such filters are used for image analysis and synthesis successfully by (Jain and Farrokhsia 199, Lee 1992). Other filter banks have also been designed for image processing by (Simoncelli etc. 1992).

The filters mentioned above are linear. Some functions are further applied to these linear filters to model the nonlinear functions of the complex cell. One way to model the complex cell is to use the power of each pair of Gabor filter $|\ (F * \mathbf{I})(x, y)\ |^2$. In fact, $|\ (F_\omega * \mathbf{I})(x, y)\ |^2$ is the local spectrum $S(\omega)$ of $\mathbf{I}$ at (x, y) smoothed by a Gaussian function. Thus it serves as a spectrum analyzer.

Although these filters are very efficient in capturing local spatial features, some problems are not well understood. For example, i) given a bank of filters, how to choose the best set of filters? especially when some of the filters are linear while others are nonlinear, or the filters are highly correlated to each other, ii) after selecting the filters, how to fuse the features captured by them into a single texture model? These questions will be answered in the rest of the paper.

2.2 MRF modeling

MRF models were popularized by Besag (Besag 1973) for modeling spatial interactions on lattice systems and were used by (Cross and Jain 1983) for texture modeling. An important characteristic of MRF modeling is that the global patterns are formed via stochastic propagation of local interactions, which is particularly appropriate for modeling textures since they are characterized by global but not predictable repetitions of similar local structures.

In MRF models, a texture is considered as a realization from a random field $\mathbf{I}$ defined over a spatial configuration $\mathcal{D}$, for example, $\mathcal{D}$ can be an array or a lattice. We denote $\mathbf{I}(\vec{v})$ as the random variable at a location $\vec{v} \in \mathcal{D}$, and let $\mathcal{N} = \{\mathcal{N}_{\vec{v}}, \vec{v} \in \mathcal{D}\}$ be a neighborhood system of $\mathcal{D}$, which is a collection of subsets of $\mathcal{D}$ satisfying 1) $\vec{v} \notin \mathcal{N}_{\vec{v}}$, and 2) $\vec{v} \in \mathcal{N}_{\vec{u}} \Longleftrightarrow \vec{u} \in \mathcal{N}_{\vec{v}}$. The pixels in $\mathcal{N}_{\vec{v}}$ are called neighbors of $\vec{v}$. A subset C of $\mathcal{D}$ is a clique if every pair of distinct pixels in C are neighbors of each other; $\mathcal{C}$ denotes the set of all cliques.

Definition. $p(\mathbf{I})$ is an *MRF distribution* with respect to $\mathcal{N}$ if $p(\mathbf{I}(\vec{v}) \mid \mathbf{I}(-\vec{v})) = p(\mathbf{I}(\vec{v}) \mid \mathbf{I}(\mathcal{N}_{\vec{v}}))$, where $\mathbf{I}(-\vec{v})$ denotes the values of all pixels other than $\vec{v}$, and for $A \subset \mathcal{D}$, $\mathbf{I}(A)$ denotes the values of all pixels in A.

Definition. $p(\mathbf{I})$ is a *Gibbs distribution* with respect to $\mathcal{N}$ if

$$p(\mathbf{I}) = \frac{1}{Z} \exp\{-\sum_{C \in \mathcal{C}} \lambda_C(\mathbf{I}(C))\}, \tag{3}$$

where Z is the normalizing constant (or partition function), and $\lambda_C()$ is a function of intensities of pixels in clique C (called potential of C). Some constraints can be imposed on λ_C for them to be uniquely determined.

The Hammersley-Clifford theorem establishes the equivalence between MRF and the Gibbs distribution (Besag 1973) :

Theorem 1 *For a given $\mathcal{N}$, $p(\mathbf{I})$ is an MRF distribution $\Longleftrightarrow$ $p(\mathbf{I})$ is a Gibbs distribution.*

This equivalence provides a general method for specifying an MRF on $\mathcal{D}$, i.e., first choose an $\mathcal{N}$, and then specify λ_C. The MRF is *stationary* if for every $C \in \mathcal{C}$, λ_C depends only on the relative positions of its pixels. This is often assumed in texture modeling.

Existing MRF models for texture modeling are mostly auto-models (Besag 1973) with pair potentials, i.e. $\lambda_C \equiv 0$ if $|C| > 2$, and $p(\mathbf{I})$ has the following form

$$p(\mathbf{I}) = \frac{1}{Z}\exp\{\sum_{\vec{v}} g(\mathbf{I}(\vec{v})) + \sum_{\vec{u},\vec{v}} \beta_{\vec{u}-\vec{v}}\mathbf{I}(\vec{u})\mathbf{I}(\vec{v})\}, \tag{4}$$

where $\beta_{-\vec{u}} = \beta_{\vec{u}}$ and $\beta_{\vec{u}-\vec{v}} \equiv 0$ unless $\vec{u}$ and $\vec{v}$ are neighbors.

The above MRF model is usually specified through conditional distributions,

$$p(\mathbf{I}(\vec{v}) \mid \mathbf{I}(-\vec{v})) \propto \exp\{g(\mathbf{I}(\vec{v}) + \sum_{\vec{u}} \beta_{\vec{v}-\vec{u}}\mathbf{I}(\vec{u})\mathbf{I}(\vec{v})\},$$

where the neighborhood is usually of order less than or equal to three pixels, and some further restrictions are usually imposed on g for $p(\mathbf{I}(\vec{v}) \mid \mathbf{I}(-\vec{v}))$ to be a linear regression or the generalized linear model.

Two commonly used auto-models are the auto-binomial model and the auto-normal model. The auto-binomial model is used for images with finite grey levels $\mathbf{I}(\vec{v}) \in \{0, 1, ..., G-1\}$ (Cross and Jain 1983), the conditional distribution is a logistic regression,

$$\mathbf{I}(\vec{v}) \mid \mathbf{I}(-\vec{v}) \sim \text{binomial}(G, p_{\vec{v}}), \tag{5}$$

where

$$\log\frac{p_{\vec{v}}}{1-p_{\vec{v}}} = \alpha + \sum_{\vec{u}} \beta_{\vec{u}-\vec{v}}\mathbf{I}(\vec{u}).$$

It can be shown that the joint distribution is of the form

$$p(\mathbf{I}) = \frac{1}{Z}\exp\{\sum_{\vec{v}}(\alpha\mathbf{I}(\vec{v}) + \log\binom{G}{\mathbf{I}(\vec{v})}) + \sum_{\vec{u},\vec{v}} \beta_{\vec{u}-\vec{v}}\mathbf{I}(\vec{u})\mathbf{I}(\vec{v})\} \tag{6}$$

When $G = 2$, the auto-binomial model reduces to the auto-logistic model (i.e., the Ising model), which is used to model binary images.

The auto-normal model is used for images with continuous grey levels (Yuan and Rao 1993). It is evident that if an MRF $p(\mathbf{I})$ is a multivariate normal distribution, then $p(\mathbf{I})$ must be auto-normal, so the auto-normal model is also called a Gaussian MRF or GMRF. The conditional distribution is a normal regression,

$$\mathbf{I}(\vec{v}) \mid \mathbf{I}(-\vec{v}) \sim N(\mu + \sum_{\vec{u}} \beta_{\vec{v}-\vec{u}}(\mathbf{I}(\vec{u}) - \mu), \sigma^2), \tag{7}$$

and $p(\mathbf{I})$ is of the form

$$p(\mathbf{I}) = \frac{1}{(2\pi\sigma^2)^{n/2}} |B|^{1/2} \exp\{-\frac{1}{2\sigma^2}(\mathbf{I} - \mu)^T B(\mathbf{I} - \mu)\}, \tag{8}$$

i.e., the multivariate normal distribution $N(\mu, \sigma^2 B^{-1})$ where the diagonal elements of B are unity and the off-diagonal $(\vec{u}, \vec{v})$ element of it is $-\beta_{\vec{u}-\vec{v}}$.

Another MRF model for texture is the ϕ-model (Geman and Graffigne 1986):

$$p(\mathbf{I}) = \frac{1}{Z} \exp\{-\sum_{<\vec{u},\vec{v}>} \lambda_{|\vec{u}-\vec{v}|} \phi(\mathbf{I}(\vec{u}) - \mathbf{I}(\vec{v}))\}, \tag{9}$$

where ϕ is a known even symmetric function, and the ϕ-model can be viewed as extended from the Potts model (Winkler 1995).

The advantage of the auto-models is that the parameters in the models can be easily inferred by auto-regression, but they are severely limited in the following two aspects: i) the cliques are too small to capture features of texture, ii) the statistics on the cliques specifies only the first-order and second order moments (e.g. means and covariances for GMRF). However, many textures has local structures much larger than three or four pixels, and the covariance information or equivalently spectrum can not adequately characterize textures, as suggested the existence of distinguishable texture pairs with identical second-order or even third-order moments, as well as indistinguishable texture pairs with different second-order moments (Diaconis and Freeman 1981). Moreover many textures are strongly non-Gaussian, regardless of neighborhood size.

The underlying cause of these limitations is that equation 3 involves too many parameters if we increase the neighborhood size or the order of the statistics, even for the simplest auto-models. This suggests that we need carefully designed functional forms for $\lambda_C()$ to efficiently characterize local interactions as well as the statistics on the local interactions.

3 From maximum entropy to FRAME model

3.1 The basics of maximum entropy

Maximum entropy (ME) is an important principle in statistics for constructing a probability distributions p on a set of random variables X (Jaynes 1957). Suppose the available information is the expectations of some known functions $\phi_n(x)$, i.e., $E_p[\phi_n(x)] = \int \phi_n(x)p(x)dx = \mu_n$ for $n = 1, ..., N$. Let Ω be the set of all probability distribution $p(x)$ which satisfy the constraints, i.e.,

$$\Omega = \{p(x) \mid E_p[\phi_n(x)] = \mu_n, n = 1, ..., N\}. \tag{10}$$

The ME principle suggests that a good choice of the probability distribution is the one that has the maximum entropy, i.e.,

$$p^*(x) = \arg\max\{-\int p(x)\log p(x)dx\}, \tag{11}$$

subject to $\quad E_p[\phi_n(x)] = \int \phi_n(x)p(x)dx = \mu_n, \quad n = 1, ..., N,$

and $\quad \int p(x)dx = 1.$

By Lagrange multipliers, the solution for $p(x)$ is:

$$p(x;\Lambda) = \frac{1}{Z(\Lambda)}\exp\{-\sum_{n=1}^{N}\lambda_n\phi_n(x)\}, \tag{12}$$

where $\Lambda = (\lambda_1, \lambda_2, ..., \lambda_n)$ is the Lagrange parameter, and $Z(\Lambda) = \int \exp\{-\sum_{n=1}^{N}\lambda_n\phi_n(x)\}dx$ is the partition function that depends on Λ and it has the following properties:

i) $$\frac{\partial \log Z}{\partial \lambda_i} = \frac{1}{Z}\frac{\partial Z}{\partial \lambda_i} = -E_{p(x;\Lambda)}[\phi_i(x)]$$

ii) $$\frac{\partial^2 \log Z}{\partial \lambda_i \partial \lambda_j} = E_{p(x;\Lambda)}[(\phi_i(x) - E_{p(x;\Lambda)}[\phi_i(x)])(\phi_j(x) - E_{p(x;\Lambda)}[\phi_j(x)])]$$

In equation (12), $(\lambda_1, ..., \lambda_N)$ is determined by the constraints in equation (11). But a closed form solution for $(\lambda_1, ..., \lambda_N)$ is not available in general, especially when $\phi_n(\cdot)$ gets complicated, so instead we seek numerical solutions by solving the following equations iteratively.

$$\frac{d\lambda_n}{dt} = E_{p(I;\Lambda)}[\phi_n(x)] - \mu_n, \qquad n = 1, 2, ..., N. \tag{13}$$

The second property of the partition function $Z(\Lambda)$ tells us that the Hessian matrix of $\log Z(\Lambda)$ is the covariance matrix of vector $(\phi_1(x), \phi_2(x), ..., \phi_N(x))$ which is definitely positive [2], and $\log Z(\Lambda)$ is strictly concave with respect to $(\lambda_1, ..., \lambda_N)$, so is $\log p(x; \Lambda)$. Therefore given a set of consistent constraints, there is a unique solution for $(\lambda_1, ..., \lambda_N)$ in equation (13).

3.2 Deriving the FRAME model

Let image $\mathbf{I}$ be defined on a discrete domain $\mathcal{D}$, $\mathcal{D}$ can be a $N \times N$ lattice. For each pixel $\vec{v} \in \mathcal{D}$, $\mathbf{I}(\vec{v}) \in \mathcal{L}$, and $\mathcal{L}$ is an interval of $\mathbf{R}$ or $\mathcal{L} \subset \mathbf{Z}$. For each texture, we assume that there exists a "true" joint probability density $f(\mathbf{I})$ over the image space $\mathcal{L}^{|\mathcal{D}|}$, and $f(\mathbf{I})$ should concentrate on a subspace of $\mathcal{L}^{|\mathcal{D}|}$ which corresponds to texture images that have perceptually similar texture appearances. Before we derive the FRAME model, we first fix the notation as below.

Given an image $\mathbf{I}$ and a filter $F^{(\alpha)}$ with $\alpha = 1, 2, ..., K$ being an index of filter, we let $\mathbf{I}^{(\alpha)}(\vec{v}) = F^{(\alpha)} * \mathbf{I}(\vec{v})$ be the filter response at location $\vec{v}$, and $\mathbf{I}^{(\alpha)}$ the filtered image. The marginal empirical distribution (histogram) of $\mathbf{I}^{(\alpha)}$ is

$$H^{(\alpha)}(z) = \frac{1}{|\mathcal{D}|} \sum_{\vec{v} \in \mathcal{D}} \delta(z - \mathbf{I}^{(\alpha)}(\vec{v})),$$

where $\delta()$ is the Dirac delta function. The marginal distribution of $f(\mathbf{I})$ with respect to $F^{(\alpha)}$ at location $\vec{v}$ is denoted by

$$f_{\vec{v}}^{(\alpha)}(z) = \int \int_{\mathbf{I}^{(\alpha)}(\vec{v}) = z} f(\mathbf{I}) d\mathbf{I} = E_f[\delta(z - \mathbf{I}^{(\alpha)}(\vec{v}))].$$

At first thought, it seems an intractable problem to estimate $f(\mathbf{I})$ due to the overwhelming dimensionality of image $\mathbf{I}$. To reduce dimensions, we first introduce the following theorem.

[2]Here, it is reasonable to assume that $\phi_n(x)$ is independent of $\phi_j(x)$ if $i \neq j$

Theorem 2 *Let $f(\mathbf{I})$ be the $\mid \mathcal{D} \mid$-dimensional continuous probability distribution of a texture, then $f(\mathbf{I})$ is a linear combination of $f^{(\xi)}$, the latter are the marginal distributions on the linear filter response $F^{(\xi)} * \mathbf{I}$.*

[**Proof**]. By inverse Fourier transform, we have

$$f(\mathbf{I}) = \frac{1}{(2\pi)^{|\mathcal{D}|}} \int \cdot \int e^{2\pi i < \mathbf{I},\ \xi >} \hat{f}(\xi) d\xi$$

where $\hat{f}(\xi)$ is the characteristic function of $f(\mathbf{I})$, and

$$\begin{aligned} \hat{f}(\xi) &= \int \cdot \int e^{-2\pi i < \xi,\ \mathbf{I} >} f(\mathbf{I}) d\mathbf{I} \\ &= \int e^{-2\pi iz} dz \int \cdot \int_{<\xi,\ \mathbf{I}>=z} f(\mathbf{I}) d\mathbf{I} \\ &= \int e^{-2\pi iz} dz \int \cdot \int \delta(z- < \xi,\ \mathbf{I} >) f(\mathbf{I}) d\mathbf{I} \\ &= \int e^{-2\pi iz}\ f^{(\xi)}(z) dz \end{aligned}$$

where $< \cdot, \cdot >$ is the inner product, and by definition $f^{(\xi)}(z) = \int \cdot \int \delta(z- < \xi,\ \mathbf{I} >) f(\mathbf{I}) d\mathbf{I}$ is the marginal distribution of $F^{(\xi)} * \mathbf{I}$, and we define $F^{(\xi)}(\vec{v}) = \xi(\vec{v})$ as a linear filter. □

Theorem 2 transforms $f(\mathbf{I})$ into a linear combination of its one dimensional marginal distributions.[3] Thus it motivates a new method for inferring $f(\mathbf{I})$: construct a distribution $p(\mathbf{I})$ so that $p(\mathbf{I})$ has the same marginal distributions $f^{(\xi)}$. If $p(\mathbf{I})$ matches all marginal distributions of $f(\mathbf{I})$, then $p(\mathbf{I}) = f(\mathbf{I})$. But this method will involve uncountable number of filters and each filter $F^{(\xi)}$ is as big as image $\mathbf{I}$.

Our second motivation comes from recent psychophysical research on human texture perception, and the latter suggests that two homogeneous textures are often difficult to discriminate when they produce similar *marginal distributions* for responses from *a bank of filters* (Bergen and Adelson 1991, Chubb and Landy 1991). This means that it is plausible to ignore some statistical properties of $f(\mathbf{I})$ which are not important for human texture discrimination.

To make texture modeling a tractable problem, in the rest of this paper we make the following assumptions to limit the number of filters and the window size of each

[3] It may help understand the spirit of this theorem by comparing it to the slice-reconstruction of 3D volume in tomography.

filter for computational reason, though these assumptions are not necessary conditions for our theory to hold true. 1). We limit our model to homogeneous textures, thus $f(\mathbf{I})$ is stationary with respect to location $\vec{v}$. [4] 2). For a given texture, all features which concern texture perception can be captured by "locally" supported filters. In other words, the sizes of filters should be smaller than the size of the image. For example, the size of image is 256×256 pixels, and the sizes of filters we used are limited to be less than 33×33 pixels. These filters can be linear or non-linear as we discussed in section (2.1). 3). Only a finite set of filters are used to estimate $f(\mathbf{I})$

Assumptions 1) and 2) are made because we often have access to only one observed (training) texture image. For a given observed image $\mathbf{I}^{obs}$ and a filter $F^{(\alpha)}$, we let $\mathbf{I}^{obs(\alpha)}$ denote the filtered image, and $H^{obs(\alpha)}(z)$ the histogram of $\mathbf{I}^{obs(\alpha)}$. According to assumption 1), $f_{\vec{v}}^{(\alpha)}(z) = f^{(\alpha)}(z)$ is independent of $\vec{v}$. By ergodicity, $H^{obs(\alpha)}(z)$ makes a consistent estimator to $f^{(\alpha)}(z)$. Assumption 2) ensures that the image size is lager relative to the support of filters, so that ergodicity takes effect for $H^{obs(\alpha)}(z)$ to be an accurate estimate of $f^{(\alpha)}(z)$.

Now for a specific texture, let $S_K = \{F^{(\alpha)}, \alpha = 1, ..., K\}$ be a finite set of well selected filters, and $f^{(\alpha)}(z), \alpha = 1, ..., K$ are the corresponding marginal distributions of $f(\mathbf{I})$. We denote the probability distribution $p(\mathbf{I})$ which matches these marginal distributions as a set,

$$\Omega_K = \{p(\mathbf{I}) \mid E_p[\delta(z - \mathbf{I}^{(\alpha)}(\vec{v}))] = f^{(\alpha)}(z) \quad \forall z \in R,\ \forall \alpha = 1, ..., K,\ \forall \vec{v} \in \mathcal{D}\}, \tag{14}$$

where $E_p[\delta(z - \mathbf{I}^{(\alpha)}(\vec{v}))]$ is the marginal distribution of $p(\mathbf{I})$ with respect to filter $F^{(\alpha)}$ at location $\vec{v}$. Thus according to assumption 3), any $p(\mathbf{I}) \in \Omega$ is perceptually a good enough model for the texture, provided that we have enough well selected filters. Then we choose from Ω a distribution $p(\mathbf{I})$ which has the maximum entropy,

$$p(\mathbf{I}) = \arg\max\{-\int p(\mathbf{I}) \log p(\mathbf{I}) d\mathbf{I}\}, \tag{15}$$

$$\begin{array}{lll} \text{subject to} & E_p[\delta(z - \mathbf{I}^{(\alpha)}(\vec{v}))] = f^{(\alpha)}(z) & \forall z \in R,\ \forall \alpha = 1, ..., K,\ \forall \vec{v} \in \mathcal{D} \\ \text{and} & \int p(\mathbf{I}) d\mathbf{I} = 1. & \end{array}$$

[4] Throughout this paper, we use circulant boundary conditions.

The reason for us to choose the maximum entropy (ME) distribution is that while $p(\mathbf{I})$ satisfies the constraints along some dimensions, it is made as random as possible in other unconstrained dimensions, since entropy is a measure of randomness. In other words, $p(\mathbf{I})$ should represent information no more than that is available. Therefore an ME distribution gives the simplest explanation for the constraints and thus the purest fusion of the extracted features.

The constraints on equation (15) differ from the ones given in section (3.1) in that z takes continuous real values, hence there are uncountable number of constraints, therefore, the Lagrange parameter λ takes the form as a function of z. Also since the constraints are the same for all locations $\vec{v} \in \mathcal{D}$, λ should be independent of $\vec{v}$. Solving this maximization problem gives the ME distribution:

$$p(\mathbf{I}; \Lambda_K, S_K) = \frac{1}{Z(\Lambda_K)} \exp\{-\sum_{\vec{v}} \sum_{\alpha=1}^{K} \int \lambda^{(\alpha)}(z) \delta(z - \mathbf{I}^{\alpha}(\vec{v})) dz\}, \quad (16)$$

$$= \frac{1}{Z(\Lambda_K)} \exp\{-\sum_{\vec{v}} \sum_{\alpha=1}^{K} \lambda^{(\alpha)}(\mathbf{I}^{(\alpha)}(\vec{v}))\}, \quad (17)$$

where $S_K = \{F^{(1)}, F^{(2)}, ..., F^{(K)}\}$ is a set of selected filters, and $\Lambda_K = (\lambda^{(1)}(), \lambda^{(2)}(), ..., \lambda^{(K)}())$ is the Lagrange parameter. Note that in equation (17), for each filter $F^{(\alpha)}$, $\lambda^{(\alpha)}()$ takes the form as a continuous function of the filter response $\mathbf{I}^{(\alpha)}(\vec{v})$.

To proceed further, let's derive a discrete form of equation (17). Assume that the filter response $\mathbf{I}^{(\alpha)}(\vec{v})$ is quantitized into L discrete grey levels, therefore z takes values from set $\{z_1^{(\alpha)}, z_2^{(\alpha)}, ..., z_L^{(\alpha)}\}$. In general, the width of these bins do not have to be equal, and the number of grey levels L for each filter response may vary. As a result, the marginal distributions and histograms are approximated by piecewisely constant functions of L bins, and we denote these piecewise functions as vectors. $H^{(\alpha)} = (H_1^{(\alpha)}, H_2^{(\alpha)}, ..., H_L^{(\alpha)})$ is the histogram of $\mathbf{I}^{(\alpha)}$, $H^{obs(\alpha)}$ denotes the histogram of $\mathbf{I}^{obs(\alpha)}$, and the potential function $\lambda^{(\alpha)}()$ is approximated by vector $\lambda^{(\alpha)} = (\lambda_1^{(\alpha)}, \lambda_2^{(\alpha)}, ..., \lambda_L^{(\alpha)})$.

So equation (16) is rewritten as:

$$p(\mathbf{I}; \Lambda_K, S_K) = \frac{1}{Z(\Lambda_K)} \exp\{-\sum_{\vec{v}} \sum_{\alpha=1}^{K} \sum_{i=1}^{L} \lambda_i^{(\alpha)} \delta(z_i^{(\alpha)} - \mathbf{I}^{(\alpha)}(\vec{v}))\},$$

by changing the order of summations:

$$p(\mathbf{I}; \Lambda_K, S_K) = \frac{1}{Z(\Lambda_K)} \exp\{-\sum_{\alpha=1}^{K}\sum_{i=1}^{L} \lambda_i^{(\alpha)} H_i^{(\alpha)}\},$$
$$= \frac{1}{Z(\Lambda_K)} \exp\{-\sum_{\alpha=1}^{K} < \lambda^{(\alpha)},\ H^{(\alpha)} >\}. \tag{18}$$

The virtue of equation (18) is that it provides us with a simple parametric model for the probability distribution on $\mathbf{I}$, and this model has the following properties,

- $p(\mathbf{I}; \Lambda_K, S_K)$ is specified by $\Lambda_K = (\lambda^{(1)}, \lambda^{(2)}, ..., \lambda^{(K)})$ and S_K.
- Given an image $\mathbf{I}$, its histograms $H^{(1)}, H^{(2)}, ..., H^{(K)}$ are sufficient statistics, i.e. $p(\mathbf{I}; \Lambda_K, S_K)$ is a function of $(H^{(1)}, H^{(2)}, ..., H^{(K)})$.

We plug equation (18) into the constraints of the ME distribution, and solve for $\lambda^{(\alpha)}, \alpha = 1, 2, ..., K$ iteratively by the following equations,

$$\frac{d\lambda^{(\alpha)}}{dt} = E_{p(\mathbf{I};\Lambda_K,S_K)}[H^{(\alpha)}] - H^{obs(\alpha)}. \tag{19}$$

In equation (19), we have substituted $H^{obs(\alpha)}$ for $f^{(\alpha)}$, and $E_{p(\mathbf{I};\Lambda_K,S_K)}(H^{(\alpha)})$ is the expected histogram of the filtered image $\mathbf{I}^{(\alpha)}$ where $\mathbf{I}$ follows $p(\mathbf{I}; \Lambda_K, S_K)$ with the current Λ_K. Equation (19) converges to the unique solution at $\Lambda_K = \hat{\Lambda}_K$ as we discussed in section (3.1), and $\hat{\Lambda}_K$ is called the ME-estimator.

It is worth mentioning that this ME-estimator is equivalent to the maximum likelihood estimator (MLE),

$$\hat{\Lambda}_K = \arg\max_{\Lambda_K} \log p(\mathbf{I}^{obs}; \Lambda_K, S_K)$$
$$= \arg\max_{\Lambda_K} -\log Z(\Lambda_K) - \sum_{\alpha=1}^{K} < \lambda^{(\alpha)},\ H^{obs(\alpha)} > . \tag{20}$$

By gradient accent, maximizing the log-likelihood gives equation (19), following property i) of the partition function $Z(\Lambda_K)$.

In Equation (19), at each step, given Λ_K and hence $p(\mathbf{I}; \Lambda_K, S_K)$, the analytic form of $E_{p(\mathbf{I};\Lambda_K,S_K)}(H^{(\alpha)})$ is not available, instead we propose the following method to estimate it as we did for $f^{(\alpha)}$ before. We draw a typical sample from $p(\mathbf{I}; \Lambda_K, S_K)$, and thus synthesize a texture image $\mathbf{I}^{syn}$. Then we use the histogram $H^{syn(\alpha)}$ of $\mathbf{I}^{syn(\alpha)}$ to approximate $E_{p(\mathbf{I};\Lambda_K,S_K)}(H^{(\alpha)})$. This requires that the size of $\mathbf{I}^{syn}$ that we are synthesizing should be large enough.[5]

To draw a typical sample image from $p(\mathbf{I};\Lambda_K,S_K)$, we use the Gibbs sampler (Geman and Geman 1984) which simulates a Markov chain in the image space $\mathcal{L}^{|\mathcal{D}|}$. The Markov chain starts from any random image, for example, a white noise image, and it converges to a stationary process with distribution $p(\mathbf{I};\Lambda_K,S_K)$. Thus when the Gibbs sampler converges, the images synthesized follow distribution $p(\mathbf{I};\Lambda_K,S_K)$.

In summary, we propose the following algorithm for inferring the underlying probability model $p(\mathbf{I};\Lambda_K,S_K)$ and for synthesizing the texture according to $p(\mathbf{I};\Lambda_K,S_K)$. The algorithm stops when the subband histograms of the synthesized texture closely match the corresponding histograms of the observed images.[6]

[5]Empirically, 128×128 or 256×256 seems to give a good estimation.

[6]We assume the histogram of each subband $\mathbf{I}^{(\alpha)}$ is normalized such that $\sum_i H_i^{(\alpha)} = 1$, therefore all the $\{\lambda_i^{(\alpha)}, i = 1, ..., L\}$ computed in this algorithm have one extra degree of freedom for each α, i.e., we can increase $\{\lambda_i^{(\alpha)}, i = 1, ..., L\}$ by a constant without changing $p(\mathbf{I};\Lambda_K,S_K)$. This constant will be absorbed by the partition function $Z(\Lambda_K)$.

Algorithm 1. The FRAME algorithm

Input a texture image $\mathbf{I}^{obs}$.
Select a group of K filters $S_K = \{F^{(1)}, F^{(2)}, ..., F^{(K)}\}$.
Compute $\{H^{obs(\alpha)}, \quad \alpha = 1, ..., K\}$.
Initialize $\lambda_i^{(\alpha)} \leftarrow 0, \quad i = 1, 2, ..., L, \quad \alpha = 1, 2, ..., K$.
Initialize $\mathbf{I}^{syn}$ as a uniform white noise texture. [7]
Repeat
  Calculate $H^{syn(\alpha)}$ $\alpha = 1, 2, ..., K$ from $\mathbf{I}^{syn}$, use it for $E_{p(\mathbf{I};\Lambda_K, S_K)}(H^{(\alpha)})$.
  Update $\lambda^{(\alpha)}$ $\alpha = 1, 2, ..., K$ by equation (19), so $p(\mathbf{I}; \Lambda_K, S_K)$ is updated.
  Apply Gibbs sampler to flip $\mathbf{I}^{syn}$ for w *sweeps* under $p(\mathbf{I}; \Lambda_K, S_K)$
Until $\frac{1}{2}\sum_i^L \mid H_i^{obs(\alpha)} - H_i^{syn(\alpha)} \mid \leq \epsilon$ for $\alpha = 1, 2, ..., K$.

Algorithm 2. The Gibbs Sampler for w sweeps

Given image $\mathbf{I}(\vec{v})$, flip_counter$\leftarrow 0$
Repeat
  Randomly pick a location $\vec{v}$ under the uniform distribution.
  For $val = 0, ..., G-1$ with G being the number of grey levels of $\mathbf{I}$
    Calculate $p(\mathbf{I}(\vec{v}) = val \mid \mathbf{I}(-\vec{v}))$ by $p(\mathbf{I}; \Lambda_K, S_K)$.
  Randomly flip $\mathbf{I}(\vec{v}) \leftarrow val$ under $p(val \mid \mathbf{I}(-\vec{v}))$.
  flip_counter $\leftarrow$ flip_counter $+ 1$
Until flip_counter$=w \times M \times N$.

In algorithm 2, to compute $p(\mathbf{I}(\vec{v}) = val \mid \mathbf{I}(-\vec{v}))$, we set $\mathbf{I}(\vec{v})$ to val, due to Markov property, we only need to compute the changes of $\mathbf{I}^{(\alpha)}$ at the neighborhood of $\vec{v}$. The size of the neighborhood is determined by the size of filter $F^{(\alpha)}$. With the updated $\mathbf{I}^{(\alpha)}$, we calculate $H^{(\alpha)}$, and the probability is normalized such that $\sum_{val=0}^{G-1} p(\mathbf{I}(\vec{v}) = val \mid \mathbf{I}(-\vec{v})) = 1$.

In the Gibbs sampler, flipping a pixel is a step of the Markov chain, and we define flipping $|\mathcal{D}|$ pixels as a sweep, where $|\mathcal{D}|$ is the size of the synthesized image. Then the overall iterative process becomes an inhomogeneous Markov chain. At the beginning of the process, $p(\mathbf{I}; \Lambda_K, S_K)$ is a "hot" uniform distribution. By updating the parameters, the process get closer and closer to the target distribution, which is much colder. So the algorithm is very much like a simulated annealing algorithm (Geyer and Thompson 1995),

[7]Note that the white noise image with uniform distribution are the samples from $p(\mathbf{I}; \Lambda_K, S_K)$ with $\lambda_i^{(\alpha)} = 0$.

which is helpful for getting around local modes of the target distribution. We refer to (Winkler 1995) for discussion of alternative sampling methods.

The computational complexity of the above algorithm is notoriously large: $O(U \times w \times |\mathcal{D}| \times G \times K \times FH \times FW)$ with U the number of updating steps for Λ_K, w the number of sweeps each time we update Λ_K, $\mathcal{D}$ the size of image $\mathbf{I}^{syn}$, G the number of grey levels of $\mathbf{I}$, K the number of filters, and FH, FW are the average window sizes of the filters. To synthesize a 128×128 texture, the typical complexity is about $50 \times 4 \times 128 \times 128 \times 8 \times 4 \times 16 \times 16 \simeq 27$ billion arithmetic operators, and takes about one day to run on a Sun-20. Therefore it is very important to choose a small set of filter which can best capture the features of the texture. We discuss how to choose filters in section (4).

3.3 A general framework

The probability distribution we derived in the last section is of the form

$$p(\mathbf{I}; \Lambda_K, S_K) = \frac{1}{Z(\Lambda_K)} \exp\{-\sum_{\vec{v}} \sum_{\alpha=1}^{K} \lambda^{(\alpha)}(\mathbf{I}^{(\alpha)}(\vec{v}))\}. \tag{21}$$

This model is significant in the following aspects.

First, the model is directly built on the features $\mathbf{I}^{(\alpha)}(\vec{v})$ extracted by a set of filters $F^{(\alpha)}$. By choosing the filters, it can easily capture the properties of the texture at multiple scales and orientations, either linear or nonlinear. Hence it is much more expressive than the cliques used in the traditional MRF models.

Second, instead of characterizing only the first and second order moments of the marginal distributions as the auto-regression MRF models did, the FRAME model includes the whole marginal distribution. Indeed, in a simplified case, if the constraints on the probability distribution are given in the form of kth-order moments instead of marginal distributions, then the functions $\lambda^{(\alpha)}(\cdot)$ in equation (21) become polynomials of order m. In such case, the complexity of the FRAME model is measured by the following two aspects: 1) the number of filters and the size of the filter, 2) the order of the moments–m. As we will see in later experiments, equation (21) enable us to model strongly non-Gaussian textures.

It is also clear to us that all existing MRF texture models can be shown as special cases of FRAME models.

Finally, if we relax the homogeneous assumption, i.e., let the marginal distribution of $\mathbf{I}^{(\alpha)}(\vec{v})$ depend on $\vec{v}$, then by specifying these marginal distributions, denoted by $f_{\vec{v}}^{(\alpha)}$, $p(\mathbf{I})$ will have the form

$$p(\mathbf{I}) = \frac{1}{Z} \exp\{-\sum_{\vec{v}} \sum_{\alpha=1}^{K} \lambda_{\vec{v}}^{(\alpha)}(\mathbf{I}^{(\alpha)})(\vec{v})\}. \tag{22}$$

This distribution is relevant in texture segmentation where $\lambda_{\vec{v}}^{(\alpha)}$ are assumed piecewise consistent with respect to $\vec{v}$, and in shape inference when $\lambda_{\vec{v}}^{(\alpha)}$ changes systematically with respect to $\vec{v}$, resulting in a slowly varying texture. We shall not study non-stationary textures in this paper.

In summary, the FRAME model incorporates and generalizes the attractive properties of the filtering theory and the random fields models, and it interprets many previous methods for texture modeling in a unified view of point.

4 Filter selection

In the last section, we build a probability model for a given texture based on a set of filters S_K. For computational reasons S_K should be chosen as small as possible, and a key factor for successful texture modeling is to choose the right set of filters that best characterizes the features of the texture being modeled. In this section, we propose a novel method for filter selection.

4.1 Design of the filter bank

To describe a wide variety of textures, we first need to design a filter bank $\mathcal{B}$. $\mathcal{B}$ can include all previously designed multi-scale filters (Daugman 1985, Simoncelli etc 1992) or wavelets (Mallat 1989, Donoho and Johnstone 1994, Coifman and Wickerhauser 1992). In this paper, we should not discuss the optimal criterion for constructing a filter bank. Throughout the experiments in this paper, we use five kinds of filters.

1. The intensity filter $\delta()$, and it captures the DC component.
2. The isotropic center-surround filters, i.e., the Laplacian of Gaussian filters. Here

we rewrite equation (1) as:

$$F(x, y \mid 0, 0, T) = const \cdot (x^2 + y^2 - T^2) e^{-\frac{x^2+y^2}{T^2}} \quad (23)$$

where $T = \sqrt{2}\sigma$ stands for the scale of the filter. We choose eight scales with $T = \sqrt{2}/2, 1, 2, 3, 4, 5, 6$, and denote these filters by $LG(T)$.

3. The Gabor filters with both sine and cosine components. We choose a simple case from equation (2):

$$Gabor(x, y \mid 0, 0, T, \theta) = const \cdot e^{\frac{1}{2T^2}(4(xcos\theta + ysin\theta)^2 + (-xsin\theta + ycos\theta)^2)} e^{-i\frac{2\pi}{T}(xcos\theta + ysin\theta)}, \quad (24)$$

We choose 6 scales $T = 2, 4, 6, 8, 10, 12$ and 6 orientations $\theta = 0^o, 30^0, 60^o, 90^o, 120^o, 150^o$. We can see that these filters are not even approximately orthogonal to each other. We denote by $Gcos(T, \theta)$ and $Gsin(T, \theta)$ the cosine and sine components of the Gabor filters.

4. The spectrum analyzers denoted by $SP(T, \theta)$, whose responses are the power of the Gabor pairs: $\mid (Gabor * \mathbf{I})(x, y) \mid^2$.

5. Some specially designed filters for one dimensional textures and the textons, see section (5) and (7).

4.2 A stepwise algorithm for filter selection

For a fixed model complexity K, one way to choose S_K from $\mathcal{B}$ is to search for all possible combinations of K filters in B and compute the corresponding $p(\mathbf{I}; \Lambda_K, S_K)$. Then by comparing the synthesized texture images following each $p(\mathbf{I}; \Lambda_K, S_K)$, we can see which set of filters is the best. Such a brute force search is computationally infeasible, and for a specific texture we often do not know what K is. Instead, we propose a stepwise greedy strategy. We start from $S_0 = \emptyset$ and hence $p(\mathbf{I}; \Lambda_0, S_0)$ an uniform distribution, and then sequentially introduce one filter at a time.

Suppose that at the k-th step we have chosen $S_k = \{F^{(1)}, F^{(2)}, ..., F^{(k)}\}$, and obtained a maximum entropy distribution

$$p(\mathbf{I}; \Lambda_k, S_k) = \frac{1}{Z(\Lambda_k)} \exp\{-\sum_{\alpha=1}^{k} < \lambda^{(\alpha)}, \ H^{(\alpha)} >\}, \quad (25)$$

so that $E_{p(\mathbf{I};\Lambda_k,S_k)}[H^{(\alpha)}] = f^{(\alpha)}$ for $\alpha = 1, 2, ..., k$. Then at the $(k+1)$-th step, for each filter $F^{(\beta)} \in \mathcal{B}/S_k$, we denote by $d(\beta) = D(E_{p(\mathbf{I};\Lambda_k,S_k)}[H^{(\beta)}], f^{(\beta)})$ the distance between $E_{p(\mathbf{I};\Lambda_k,S_k)}[H^{(\beta)}]$ and $f^{(\beta)}$, which are respectively the marginal distributions of $p(\mathbf{I};\Lambda_k,S_k)$ and $f(\mathbf{I})$ with respect to filter $F^{(\beta)}$. Intuitively, the bigger $d(\beta)$ is, the more information $F^{(\beta)}$ carries, since it reports a big difference between $p(\mathbf{I};\Lambda_k,S_k)$ and $f(\mathbf{I})$. Therefore we should choose the filter which has the maximal distance, i.e.,

$$F^{(k+1)} = \arg \max_{F^{(\beta)} \in \mathcal{B}/S_k} D(E_{p(\mathbf{I};\Lambda_k,S_k)}[H^{(\beta)}], f^{(\beta)}). \tag{26}$$

Empirically we choose to measure the distance $d(\beta)$ in terms of L_p-norm, i.e.,

$$F^{(k+1)} = \arg \max_{F^{(\beta)} \in \mathcal{B}/S_k} \frac{1}{2} \mid f^{(\beta)} - E_{p(\mathbf{I};\Lambda_k,S_k)}[H^{(\beta)}] \mid_p . \tag{27}$$

In the experiments of this paper, we choose $p = 1$.

To estimate $f^{(\beta)}$ and $E_{p(\mathbf{I};\Lambda_k,S_k)}[H^{(\beta)}]$, we applied $F^{(\beta)}$ to the observed image $\mathbf{I}^{obs}$ and the synthesized image $\mathbf{I}^{syn}$ sampled from the $p(\mathbf{I};\Lambda_k,S_k)$, and substitute the histograms of the filtered images for $f^{(\beta)}$ and $E_{p(\mathbf{I};\Lambda_k,S_k)}[H^{(\beta)}]$, i.e.,

$$F^{(k+1)} = \arg \max_{F^{(\beta)} \in \mathcal{B}/S_k} \frac{1}{2} \mid H^{obs(\beta)} - H^{syn(\beta)} \mid . \tag{28}$$

The filter selection procedure stops when the $d(\beta)$ for all filters $F^{(\beta)}$ in the filter bank are smaller than a constant ϵ. Due to fluctuation, various patches of the same observed texture image often have a certain amount of histogram variance, and we use such a variance for ϵ.

In summary, we propose the following algorithm for filter selection.

Algorithm 3. Filter Selection

Let $\mathcal{B}$ be a bank of filters, S the set of selected filters, $\mathbf{I}^{obs}$ the observed texture image, and $\mathbf{I}^{syn}$ the synthesized texture image.

Initialize $k = 0$, $S \leftarrow \emptyset$, $p(\mathbf{I}) \leftarrow$ uniform dist. $\mathbf{I}^{syn} \leftarrow$ uniform noise.
For $\alpha = 1, ..., |\mathcal{B}|$ do
  Compute $\mathbf{I}^{obs(\alpha)}$ by applying $F^{(\alpha)}$ to $\mathbf{I}^{obs}$.
  Compute histogram $H^{obs(\alpha)}$ of $\mathbf{I}^{obs(\alpha)}$.
Repeat
  For each $F^{(\beta)} \in \mathcal{B}/S$ do

Compute $\mathbf{I}^{syn(\beta)}$ by applying $F^{(\beta)}$ to $\mathbf{I}^{syn}$

Compute histogram $H^{syn(\beta)}$ of $\mathbf{I}^{syn(\beta)}$

$d(\beta) = \frac{1}{2} \mid H^{obs(\beta)} - H^{syn(\beta)} \mid$, [8]

Choose $F^{(k+1)}$ so that $d(k+1) = \max\{d(\beta) \ : \ \forall F^{(\beta)} \in \mathcal{B}/S\}$

$S \leftarrow S \cup \{F^{(k+1)}\}$, $k \leftarrow k+1$.

Starting from $p(\mathbf{I})$ and $\mathbf{I}^{syn}$, run algorithm 1 to compute new $p^*(\mathbf{I})$ and $\mathbf{I}^{syn*}$.

$p(\mathbf{I}) \leftarrow p^*(\mathbf{I})$ and $\mathbf{I}^{syn} \leftarrow \mathbf{I}^{syn*}$.

Until $d^{(\beta)} < \epsilon$

Before we conclude this section, we would like to mention that the above criterion for filter selection is related to the minimax entropy principle studied in (Zhu, Wu, and Mumford 1996). The minimax entropy principle suggests that the optimal set of filters S_K should be chosen to minimize the Kullback-Leibler distance between $p(\mathbf{I}; \Lambda_K, S_K)$ and $f(\mathbf{I})$, and the latter is measured by the entropy of the model $p(\mathbf{I}; \Lambda_K, S_K)$ up to a constant. Thus at each step $k+1$ a filter is selected so that it minimizes the entropy of $p(\mathbf{I}; \Lambda_k, S_k)$ by gradient descent, i.e.,

$$F^{(k+1)} = \arg \max_{F^{(\beta)} \in \mathcal{B}/\mathcal{B}_k} \text{entropy}(p(\mathbf{I}; \Lambda_k, S_k)) - \text{entropy}(p(\mathbf{I}; \Lambda_+, S_+))$$

where $S_+ = S_k \cup \{F^{(\beta)}\}$ and Λ_+ is the new Lagrange parameter. A brief derivation is given in the appendix.

5 Experiments in one dimension

In this section we illustrate some important concepts of the FRAME model by studying a few typical examples for 1D texture modeling. In these experiments, the filters are chosen by hand.

For one-dimensional texture the domain is a discrete array $\mathcal{D} = [0, 255]$, and a pixel is indexed by x instead of $\vec{v}$. For any $x \in [0, 255]$, $\mathbf{I}(x)$ is discretized into G grey levels, with $G = 16$ in experiment I and III, and $G = 64$ in experiment II.

Experiment I. This experiment is designed to show the analogy between filters in texture modeling and vocabulary in language description, and to demonstrate how a

[8] Since both histograms are normalized to have $sum = 1$, then $error \in [0, 1]$. We note this measure is robust with respect to the choice of the bin number L (e.q. we can take $L = 16, 32, 64$), as well as the normalization of the filters.

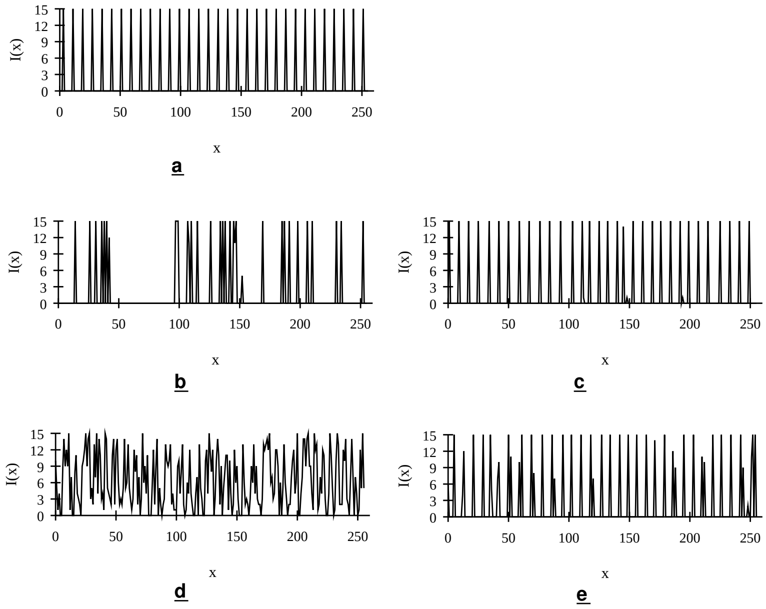


Figure 1 The observed and synthesized pulse textures. a) the observed; b) synthesized using only the intensity filter; c) intensity filter plus rectangular filter with $T = 8$; d) Gabor filter with $T = 8$; and e) Gabor filter plus intensity filter.

texture can be specified by the marginal distributions of a few well selected filters.

The observed texture, as shown in figure (1.a), is a periodic pulse signal with period $T = 8$, i.e., $\mathbf{I}(x) = 15$ once every 8 pixels and $\mathbf{I}(x) = 0$ for all the other pixels. First we choose an intensity filter, and the filter response is the signal itself. The synthesized texture by FRAME is displayed in figure (1.b). Obviously it has almost the same number of pulses as the observed one, and so has approximately the same marginal distribution for intensity. Unlike the observed texture, however, these pulses are not arranged periodically.

To capture the period of the signal, we add one more special filter, an 8×1 rectangular filter: $[1, 1, 1, 1, 1, 1, 1, 1]$, and the synthesized signal is shown in figure (1.c), which has almost the same appearance as in Figure (1.a). We can say that the probability $p(\mathbf{I})$ specified by these two filters models the properties of the input signal very well.

Figure (1.d) is the synthesized texture using a nonlinear filter which is an 1D spectrum analyzer $SP(T)$ with $T = 8$. Since the original periodic signal has flat power spectrum, and the Gabor filters only extract information in one frequency band, therefore the texture synthesized under $p(\mathbf{I})$ has power spectrum near frequency $\frac{2\pi}{8}$ but are totally free at other bands. Due to the maximum entropy principle, the FRAME model allows for the unconstrained frequency bands to be as noisy as possible. This explains why figure (1.d) is noise like while having roughly a period of $T = 8$. If we add the intensity filter, then probability $p(\mathbf{I})$ captures the observed signal again, and a synthesized texture is shown in figure (1.e).

This experiment shows that the more filters we use, the closer we can match the synthesized images to the observed. But there are some disadvantages for using too many filters. Firstly it is computationally expensive, and secondly, since we have few observed examples, it may overly constrain the probability $p(\mathbf{I})$, i.e. it may make $p(\mathbf{I})$ 'colder' than it should be.

Experiment II. In this second experiment we compare FRAME against Gaussian MRF models by showing the inadequacy of the GMRF model to express high order statistics.

To begin with, we choose a gradient filter ∇ with impulse response $[-1, 1]$ for comparison, and the filtered image is denoted by $\nabla \mathbf{I}$.

The GMRF models are concerned with only the mean and variance of the filter responses. As an example, we put the following two constraints on distribution $p(\mathbf{I})$,

$$E_p[\nabla \mathbf{I}(x)] = 0 \text{ and } E_p[\nabla \mathbf{I}(x)^2] = 12.0 \quad \forall x \in \mathcal{D}.$$

Since we use a circulant boundary, the first constraint always holds, and the resulting maximum entropy probability is

$$P(\mathbf{I}) = \frac{1}{Z} \exp\{-\lambda \sum_x (\mathbf{I}(x+1) - \mathbf{I}(x))^2\}.$$

The numeric solution given by the FRAME algorithm is $\lambda = 0.40$, and two synthesized texture images are shown in figure (3.b) and (3.c). Figure (3.a) is a white noise texture for comparison.

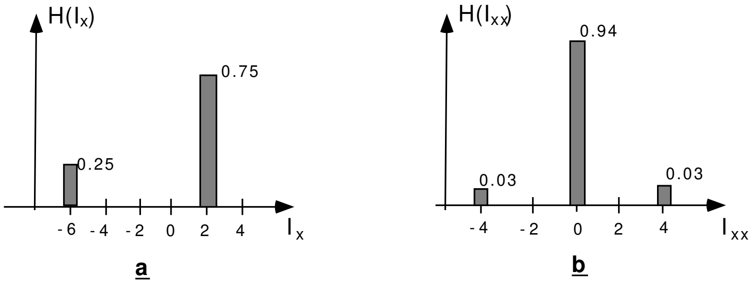


Figure 2 a) The designed marginal distribution of $\nabla \mathbf{I}$; and b) The designed marginal distribution of $\Delta \mathbf{I}$.

As a comparison, we now ask $\nabla \mathbf{I}(x)$ to follow the distribution shown in figure (2.a). Clearly in this case $E_p[\nabla \mathbf{I}(x)]$ is a non-Gaussian distribution with first and second moments as before, i.e., $mean = 0$ and $variance = 12.0$. Two synthesized textures are displayed in figure (3.d and e). The textures in figure (3.d and e) possess the same first and second order moments as in figure (3.b and c), but figure (3.d and e) have specific higher order statistics and looks more specific than in figure (3.b and c). It demonstrates that the FRAME model has more expressive power than the GMRF model.

Now we add a Laplacian filter Δ with impulse response $[0.5, -1.0, 0.5]$, and we ask $\Delta \mathbf{I}(x)$ to follow the distribution shown in figure (2.b). Clearly the number of peaks and valleys in $\mathbf{I}(x)$ are specified by the two short peaks in figure (2.b), the synthesized texture

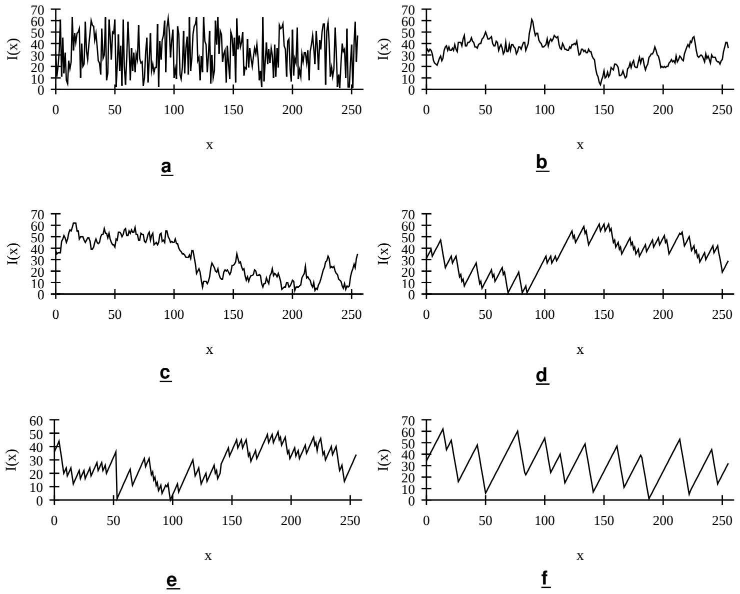


Figure 3 a. The uniform white noise texture, b.c. the texture of GMRF, d,e, the texture with higher order statistics, f. the texture specified with one more filter.

is displayed in figure (3.f). This experiment also shows the analogy between filters and vocabulary.

Experiment III. This experiment is designed to demonstrate how a single non-linear Gabor filter is capable of forming global periodic textures. The observed texture is a perfect sine wave with period $T_1 = 16$, hence it has a single Fourier component. We choose the spectrum analyzer $SP(T)$ with period $T = 16$. The synthesized texture is in figure (4.a). The same is done for another sine wave that has period $T_2 = 32$, and correspondingly the result is shown in figure (4.b). Figure (4) show clear globally periodic signals formed by single local filters. The noise is due to the frequency resolution of the filters. Since the input textures are exactly periodic, the optimal resolution will requires the Gabor filters to be as long as the input signal, which is computationally more expensive.

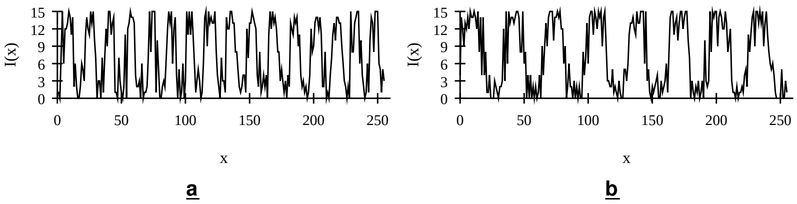


Figure 4 The observed textures are the pure sine waves with period T=16, and 32 respectively. Periodic texture synthesized by a pair of Gabor filters, a. T=16, b. T=32.

6 Experiments in two dimensions

In this section, we discuss texture modeling experiments in two dimensions. We first take one texture as an example to show in detail the procedure of algorithm 3, then we will apply algorithm 3 to other textures.

Figure (5.a) is the observed image of animal fur. We start from the uniform noise image in figure (5.b). The first filter picked by the algorithm is a Laplacian of Gaussian filter $LG(1.0)$ and its window size is 5×5. It has the largest error ($d(\beta) = 0.611$) among all the filters in the filters bank. Then we synthesize texture as shown in figure (5.c), which has almost the same histogram at the subband of this filter (the error $d(\beta)$ drops to 0.035).

Comparing figure (5.c) with figure (5.b), we notice that this filter captures local smoothness feature of the observed texture. Then the algorithm sequentially picks 5 more filters. They are 2) $Gcos(6.0, 120^o)$, 3) $Gcos(2.0, 30^o)$, 4) $Gcos(12.0, 60^o)$, 5) $Gcos(10.0, 120^o)$, 6) intensity filter $\delta()$, each of which captures features at various scales and orientations. The sequential conditional errors for these filters are respectively $0.424, 0.207, 0.132, 0.157, 0.059$ and the texture images synthesized using $k = 2, 3, 6$ filters are shown in figure (5.d,e,f). Obviously, with more filters added, the synthesized texture gets closer to the observed one.

To show more details, we display the subband images of the 6 filters in figure (6), the histograms of these subbands $H^{(\alpha)}$ and the corresponding estimated parameters $\lambda^{(\alpha)}$ are plotted in figure (7) and figure (8) respectively.

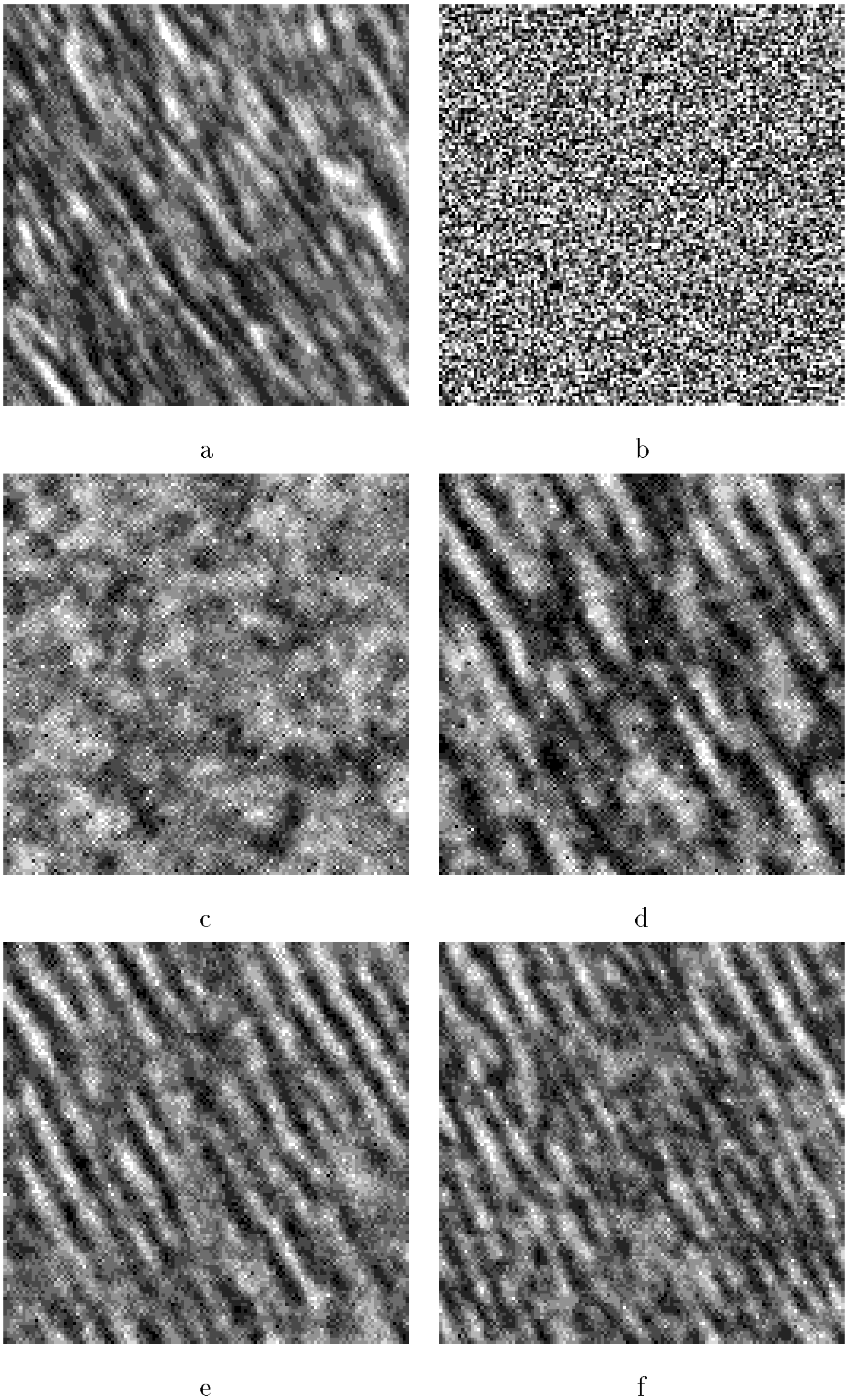


Figure 5 Synthesis of the fur texture. a is the observed texture, b,c,d,e,f are the synthesized textures using $K = 0, 1, 2, 3, 6$ filters respectively. See text for interpretation.

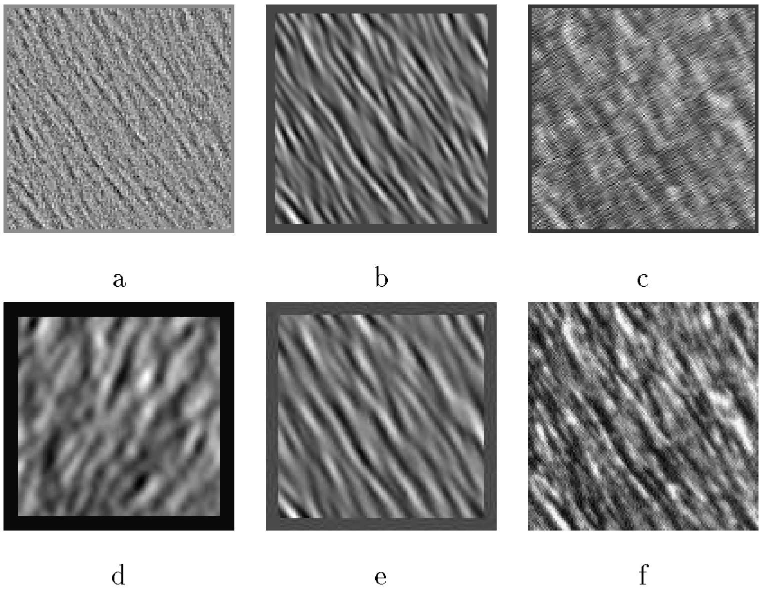


Figure 6 The subband images by applying the 6 filters to the fur image: a Laplacian of Gaussian ($T = 1.0$), b Gabor cosine ($T = 6.0, \theta = 120^o$), c Gabor cosine ($T = 2.0, \theta = 30^o$), d Gabor cosine ($T = 12, \theta = 60$), e Gabor cosine ($T = 10.0, \theta = 120^o$), f intensity filter.

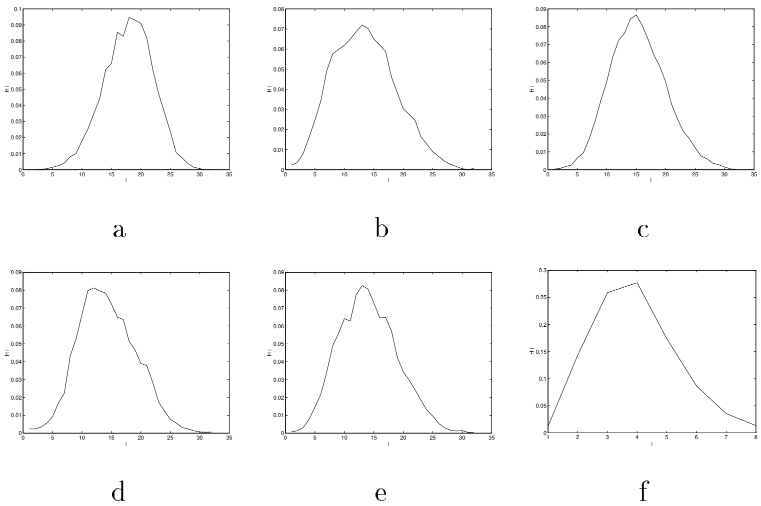


Figure 7 a,b,c,d,e,f are respectively the histograms $H^{(\alpha)}$ for $\alpha = 1, 2, 3, 4, 5, 6$

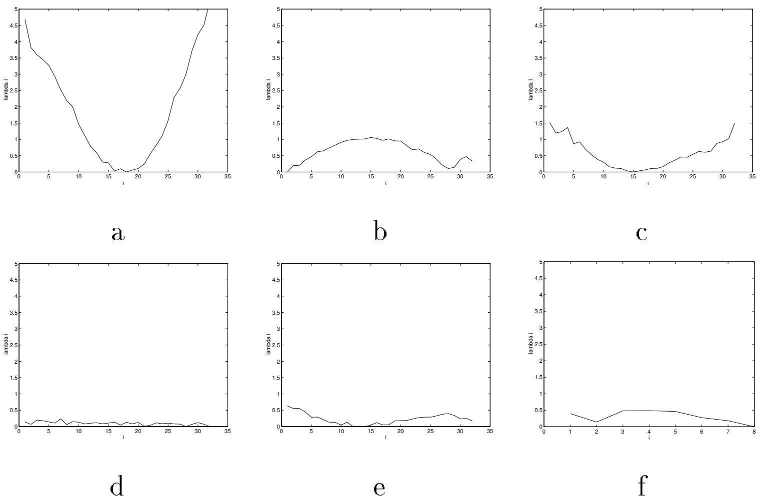
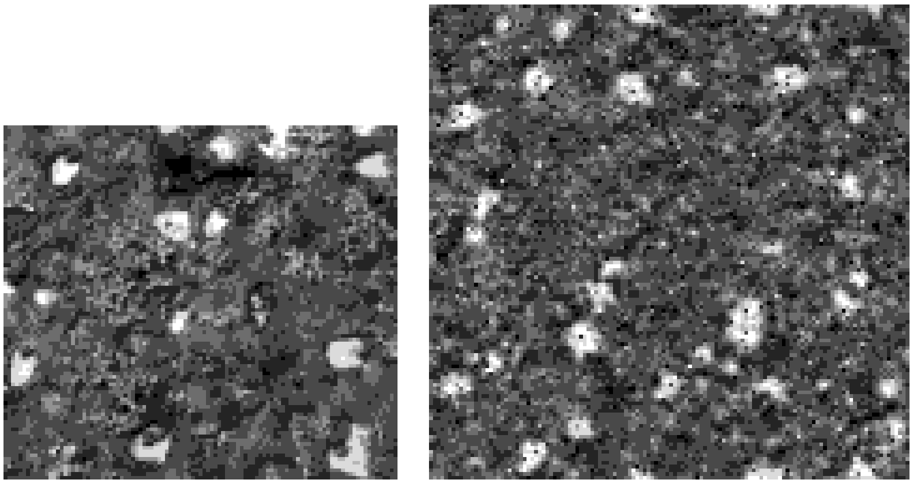


Figure 8 a,b,c,d,e,f are respectively the $\lambda^{(\alpha)}$ for $\alpha = 1, 2, 3, 4, 5, 6$

In figure (7), the histograms are approximately Gaussian functions, and correspondently, the estimated $\lambda^{(\alpha)}$ in figure (8) are close to quadratic functions. Hence in this example, the high order moments seemly do not play a major role, and the probability model can be made simpler. But this will not be always true for other textures. In figure (8), we also notice that the computed $\lambda^{(\alpha)}$ becomes smaller and smaller when α gets bigger, which suggests that the filters chosen in later steps make less and less contribution to $p(\mathbf{I})$, and thus confirms our early assumption that the marginal distributions of a small number of filtered images are good enough to capture the underlying probability distribution $f(\mathbf{I})$.

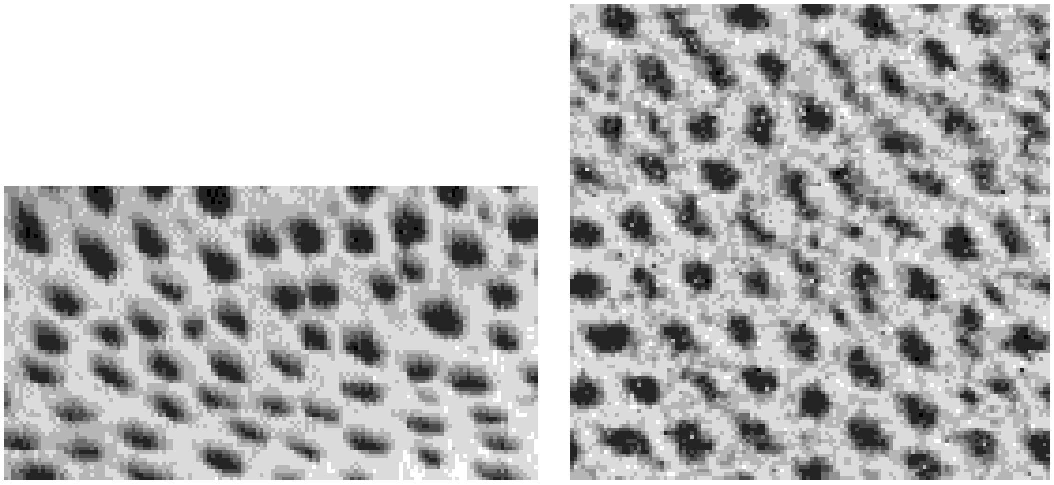
Figure (9.a) is the scene of the mud ground with footprints of animals, these footprints are filled with water and get brighter. This is a case of sparse features. Figure (9.b) is the synthesized texture using 5 filters chosen by algorithm 3.

Figure (10.a) is an image taken from the skin of cheetah. the synthesized texture using 6 filters is displayed in figure (10.b). We notice that in figure (10.a) the texture is not homogeneous, the shapes of the blobs vary with spatial locations and the left upper corner is darker than the right lower one. The synthesized texture, shown in figure (10.b), also has elongated blobs introduced by different filters, but we notice that the bright pixels



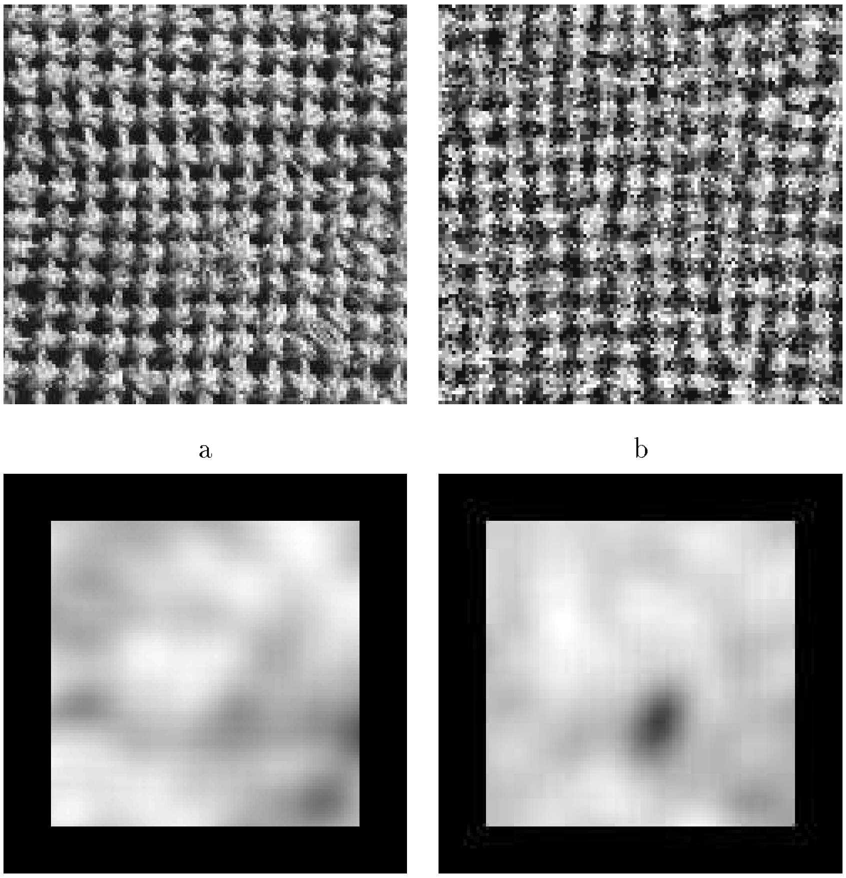
a b

Figure 9 a. the observed texture–mud, b, the synthesized one using 5 filters



a b

Figure 10 a. the observed texture–cheetah blob, b, the synthesized one using 6 filters



a b

c d

Figure 11 a. the input image of fabric, b. the synthesized image with two spectrum analyzers plus the Laplacian of Gaussian filter. c,d The filter response of the two spectrum analyzers for the fabric texture spread uniformly across the image.

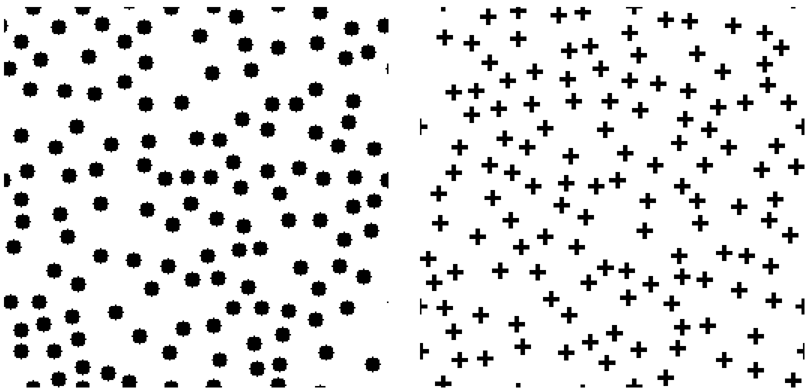
Finally we show a texture of fabric in figure (11.a), which has clear periods along both horizontal and vertical directions. We want to use this texture to test the use of non-linear filters, so we choose two spectrum analyzers to capture the first two salient periods, one in the horizontal direction, the other in the vertical direction. The filter responses $\mathbf{I}^{(\alpha)}$ $\alpha = 1, 2$, are the sum of squares of the sine and cosine component responses. The filter responses are shown in figure (11c, d), and are almost constant. We also use the intensity filter and the Laplacian of Gaussian filter $LG(\sqrt{2}/2)$ (with window size 3×3) to take care of the intensity histogram and the smoothness. The synthesized texture is displayed in figure (11.b). If we carefully look at figure (11.b), we can see that this synthesized texture has mis-arranged lines at two places, which may indicate that the sampling process was trapped in a local maximum of $p(\mathbf{I})$.

7 The Sampling strategy for textons

In this section, we study a special class of textures formed from identical textons, which psychophysicists studied extensively. Such texton images are considered as rising from a different mechanism from other textures in both psychology perception and previous texture modeling, and the purpose of this section is to demonstrate that they can still be modeled by the FRAME model, and to show an annealing strategy for computing $p(\mathbf{I}; \Lambda_K, S_K)$.

Figure (12.a,b) are two binary ($-1, +1$ for black and white pixels) texton images with circle and cross as the primitives. These two image are simply generated by sequentially superimposing 128 15×15 masks on a 256×256 lattice using uniform distribution, provided that the dropping of one mask does not destroy the existing primitives. At the center of the mask is a circle (or a cross).

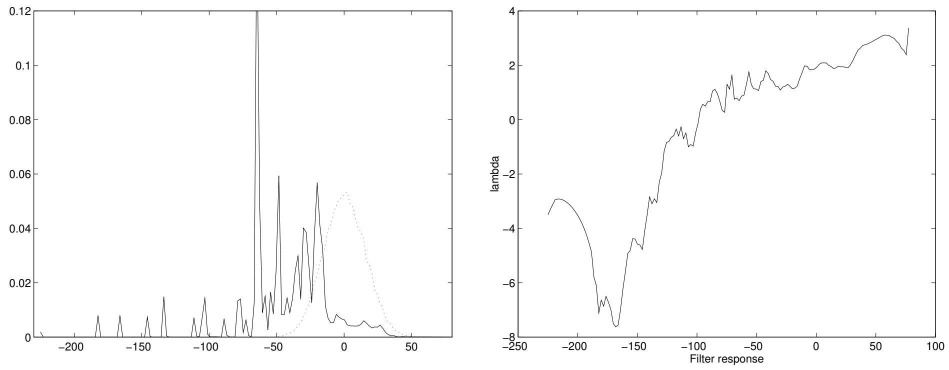
For these textures, choosing filters seems easy: we simply select the above 15×15 mask as the linear filter. Take the circle texton as an example. By applying the filter to the circle image and a uniform noise image, we obtain the histograms H^{obs} (solid curve) and $H(x)$ (dotted curve) plotted in figure (13.a). We observe that there are many isolated



a b

Figure 12 Two typical texton images. a circle, b cross

peaks in H^{obs}, which set up " potential wells" so that it becomes extremely unlikely to change a filter response at a certain location from one peak to another by flipping one pixel at a time.



a b

Figure 13 a. The solid curve is the histogram of the circle image, and the dotted curve is the histogram of the noise image; b. the estimated $\lambda()$ function in the probability model for the image of circles

To facilitate the matching process, we propose the following heuristics. We smooth H^{obs} with a Gaussian window G_σ, or equivalently run the "heat" diffusion equation on $H^{obs}(x,t)$ within the interval $[x_0, x_N]$, where x_0 and x_N are respectively the minimal and maximal filter response.

$$\frac{dH^{obs}(x,t)}{dt} = \frac{\partial^2 H^{obs}(x,t)}{\partial^2 x},$$
$$H^{obs}(x,0) = H^{obs}(x), \quad \frac{\partial H^{obs}}{\partial x}(x_0) = 0, \quad \frac{\partial H^{obs}}{\partial x}(x_N) = 0,$$

The boundary conditions help to preserve the total "heat". Obviously, the larger t is, the smoother the $H^{obs}(x,t)$ will be. Therefore we start from matching $H(x)$ to $H^{obs}(x,t)$ with a large t (see figure (14.a)), then gradually decrease t and match $H(x)$ to the histograms shown in figure (14.b,c,d,e,f) sequentially. This process is similar to the simulated annealing method. The intuitive idea is to set up "bridges" between the peaks in the original histogram, which encourages the filter response change to the two ends, where the texton forms, then we gradually destruct these "bridges".

At the end of the process, the estimated λ function for the circle texton is shown in figure (13.b), and the synthesized images are shown in figure (15). We notice that the cross texton is more difficult to deal with because it has slightly more complex structures than the circle, and may need more carefully designed filters.

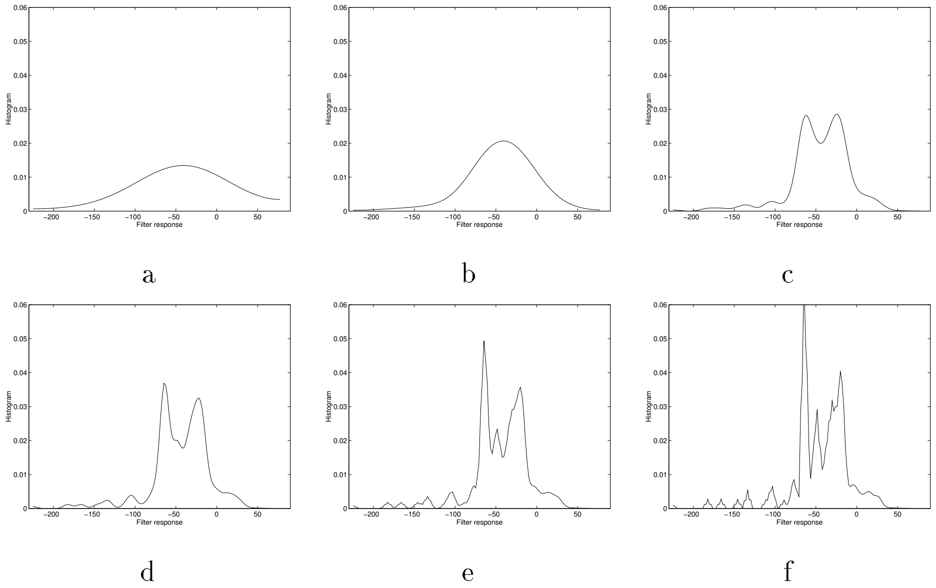


Figure 14 The diffused histogram $H^{obs}(x,t)$ with t get smaller and smaller from a to f.

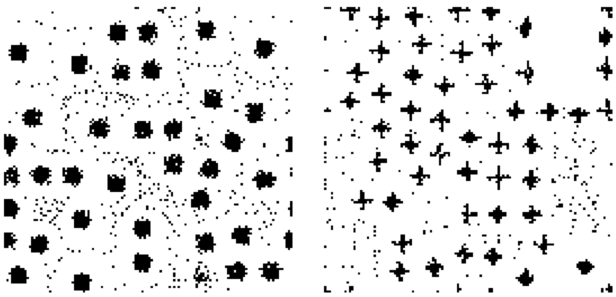


Figure 15 Two synthesized texton images.

8 Discussion

Although there is a close relationship between FRAME and the previous MRF models, the underlying philosophies are quite different. Traditional MRF approaches favor the specification of conditional distributions (Besag 1973). For auto-models, $p(\mathbf{I}(\vec{v}) \mid \mathbf{I}(-\vec{v}))$ are linear regressions or logistic regressions, so the modeling, inference, and interpretation can be done in a traditional way. While it is computationally efficient for estimating the β coefficients, this method actually limits our imagination for building a general model. Since the only way to generalize auto-models in the conditional distribution framework is to either increase neighborhood size, and thus introduce more explanatory variables in these auto-regressions, or introduce interaction terms (i.e., high order product terms of the explanatory variables). However, even with a modest neighborhood (e.g., 13×13), the parameter size will be too large for any sensible inference.

Our FRAME model, on the contrary, favors the specification of the joint distribution and characterizes local interactions by introducing non-linear functions of filter responses. This is not restricted by the neighborhood size since every filter introduces the same number of parameters regardless of its size, which enables us to explore structures at large scales (e.g., 33×33 for the fabric texture). Moreover, FRAME can easily incorporate local interactions at different scales and orientations.

It is also helpful to appreciate the difference between FRAME and the Gibbs distribution although both focus on the joint distributions. The Gibbs distribution is specified via potentials of various *cliques*, and the fact that most physical systems only have pair potentials (i.e., no potentials from the cliques with more than two pixels) is another reason why most MRF models for textures are restricted to auto-models. FRAME , on the other hand, builds potentials from finite-support filters and emphasizes the marginal distributions of filter responses.

Although it may take a large number of filters to model a wide variety of textures, when it comes to modeling a certain texture, only a parsimonious set of the most meaningful filters needs to be selected. This selectivity greatly reduces the parameter size, thus allows accurate inference and modest computing. So FRAME is like a language: it has an efficient vocabulary (of filters) capable of describing most entities (textures), and when it comes to a specific entity, a few of the most meaningful words (filters) can be selected from the vocabulary for description. This is similar to the visual coding theory (Barlow et al 1989, Field 1989) which suggests that the sparse coding scheme has advantages over the compact coding scheme. The former assumes non-Gaussian distributions for $f(\mathbf{I})$, whereas the latter assumes Gaussian distributions.

Compared to the filtering method, FRAME has the following advantages: 1) solid statistical modeling, 2) it does not rely on the reversibility or reconstruction of $\mathbf{I}$ from $\{\mathbf{I}^{(\alpha)}\}$, and thus the filters can be designed freely. For example, we can use both linear and nonlinear filters, and the filters can be highly correlated to each other, whereas in the filtering method, a major concern is whether the filters form a tight frame (Daubechies 1992).

There are various classifications for textures with respect to various attributes, such as Fourier and non-Fourier corresponding to whether the textures show periodic appearance; deterministic and stochastic corresponding to whether the textures can be characterized by some primitives and placement rules; and macro- and micro-textures in relation to the scales of local structures. FRAME erases these artificial boundaries and characterizes them in a unified model with different filters and parameter values. It has been well recognized that the traditional MRF models, as special cases of FRAME, can be used to model stochastic, non-Fourier micro-textures. From the textures we synthesized, it is evident that FRAME is also capable of modeling periodic and deterministic textures (fabric and pulses), textures with large scale elements (fur and cheetah blob), and textures with distinguishable textons (circles and cross bars), thus it realizes the full potential of MRF models.

But the FRAME model is computationally very expensive. The computational complexity of the FRAME model comes from two major aspects. I). When bigger filters are adopted to characterize low resolution features, the computational cost will increase proportionally with the size of the filter window. II). The marginal distributions $E_p[H^{(\alpha)}]$ are estimated from sampled images, which requires long iterations for high accuracy of estimation. One promising way to reduce the computational cost is to combine the pyramid representation with the pseudo-likelihood estimation (Besag 1977). The former cuts the size of low resolution filters by putting them at the high levels of the pyramid as did in (Popat and Picard 1993), and the latter approximates $E_p[H^{(\alpha)}]$ by pseudo-likelihood and thus avoid the sampling process. But this method shall not be studied in this paper.

No doubt many textures will not be easy to model, for example some human synthesized textures, such as textures on oriental rugs and clothes. It seems that the synthesis of such textures requires far more sophisticated or high-level features than those we used in this paper, and these high-level features may correspond to high-level visual process. At the same time, many theoretical issues remain yet to be fully understood, for example, the convergence properties of the sampling process and the definition of the best sampling procedures; the relationship between the sampling process and the physical process which forms the textures of nature and so on; and how to apply this texture model to the image segmentation problem (Zhu and Yuille 1996). It is our hope that this work will simulate future research efforts in this direction.

Appendix. Filter pursuit and minimax entropy.

This appendix briefly demonstrates the relationship between the filter pursuit method and the minimax entropy principle (Zhu, Wu and Mumford 1996).

Let $p(\mathbf{I}; \Lambda_K, S_K)$ be the maximum entropy distribution obtained at step k (see equation (18)), since our goal is to estimate the underlying distribution $f(\mathbf{I})$, the goodness of $p(\mathbf{I}; \Lambda_K, S_K)$ can be measured by the Kullback-Leibler distance between $p(\mathbf{I};\Lambda_K,S_K)$ and $f(\mathbf{I})$ (Kullback and Leibler 1951):

$$KL(f(\mathbf{I}),p(\mathbf{I};\Lambda_K,S_K)) = \int f(\mathbf{I})\log\frac{f(\mathbf{I})}{p(\mathbf{I};\Lambda_K,S_K)}d\mathbf{I} = E_f[\log f(\mathbf{I})] - E_f[\log p(\mathbf{I};\Lambda_K,S_K)].$$

Since $E_{p(\mathbf{I};\Lambda_K,S_K)}[H^{(\alpha)}] = E_f[H^{(\alpha)}]$ for $\alpha = 1,2,...,K$, it can be shown that $E_f[\log p(\mathbf{I};\Lambda_K,S_K)] = E_{p(\mathbf{I};\Lambda_K,S_K)}[\log p(\mathbf{I};\Lambda_K,S_K)] = -\text{entropy}(p(\mathbf{I};\Lambda_K,S_K))$, thus

$$KL(f(\mathbf{I}),p(\mathbf{I};\Lambda_K,S_K)) = \text{entropy}(p(\mathbf{I};\Lambda_K,S_K)) - \text{entropy}(f(\mathbf{I})).$$

As entropy$(f(\mathbf{I}))$ is fixed, to minimize $KL(f,p(\mathbf{I};\Lambda_K,S_K))$ we need to choose S_K such that $p(\mathbf{I};\Lambda_K,S_K)$ has the minimum entropy, while given the selected filter set S_K, $p(\mathbf{I};\Lambda_K,S_K)$ is computed by maximizing entropy$(p(\mathbf{I}))$. In other words, for a fixed filter number K, the best set of filters is chosen by

$$S_K = \arg\min_{S_K\subset\mathcal{B}}\{\max_{p\in\Omega_K}\text{entropy}(p(\mathbf{I}))\} \tag{29}$$

where Ω_K is defined as equation (14). We call equation (29) the *minimax entropy principle* (Zhu, Wu, Mumford 1996).

A stepwise greedy algorithm to minimize the entropy proceeds as the following. At step $k+1$, suppose we choose $F^{(\beta)}$, and obtain the ME distribution $p(\mathbf{I};\Lambda_+,S_+)$ so that $E_{p(\mathbf{I};\Lambda_+,S_+)}[H^{(\alpha)}] = f^{(\alpha)}$ for $\alpha = 1,2,..,k,\beta$. Then the goodness of $F^{(\beta)}$ is measured by the decrease of the Kullback-Leibler distance $KL(f(\mathbf{I}),p(\mathbf{I};\Lambda_k,S_k)) - KL(f(\mathbf{I}),p(\mathbf{I};\Lambda_+,S_+))$. It can be shown that

$$KL(f(\mathbf{I}),p(\mathbf{I};\Lambda_k,S_k)) - KL(f(\mathbf{I}),p(\mathbf{I};\Lambda_+,S_+)) \tag{30}$$

$$= \frac{1}{2}(f^{(\beta)} - E_{p(\mathbf{I};\Lambda_k,S_k))}[H^{(\beta)}])^T M^{-1}(f^{(\beta)} - E_{p(\mathbf{I};\Lambda_k,S_k))}[H^{(\beta)}]), \tag{31}$$

where M is a covariance matrix of $H^{(\beta)}$, for details see (Zhu,Wu, Mumford 1996). Equation (31) measures a distance between $f^{(\beta)}$ and $E_{p(\mathbf{I};\Lambda_k,S_k)}[H^{(\beta)}]$ in terms of variance, and therefore suggests a new form for the distance $D(E_{p(\mathbf{I};\Lambda_k,S_k))}[H^{(\beta)}],f^{(\beta)})$ in equation (26), and this new form emphasizes the tails of the marginal distribution where important texture features lies, but the computational complexity is higher than the L_1-norm distance. So far we have shown the filter selection in algorithm 3 is closely related to a minimax entropy principle.

Acknowledgments

This work was supported by the NSF grant DMS-91-21266 to David Mumford. The second author was supported by a grant to D.B. Rubin.

References

[1] Barlow, H.B., Kaushal, T.P. and Mitchison, G.J., "Finding minimum entropy codes." *Neural Computation.* Vol. 1, pp412-423, 1989.

[2] Bergen, J. R. and Adelson, E. H. "Theories of visual texture perception." *Spatial Vision* Regan, D. (eds.), CRC press, 1991.

[3] Besag, J. "Spatial interaction and the statistical analysis of lattice systems (with discussion)." *J. Royal Stat. Soc., B*, **36**, 192-236. 1973.

[4] Besag, J. 1977. "Efficiency of pseudolikelihood estimation for simple Gaussian fields." Biometrika. **64**, 616-618.

[5] Chubb, C. and Landy, M. S. "Orthogonal distribution analysis: a new approach to the study of texture perception." *Comp. Models of Visual Proc.* Landy. etc (ed.) MIT press 1991.

[6] Coifman, R.R. and Wickerhauser, M.V.. "Entropy based algorithms for best basis selection." *IEEE Trans. on Information Theory.*, Vol.38, pp713-718, 1992.

[7] Cross, G. R. and Jain, A. K. "Markov random field texture models." *IEEE, PAMI*, **5**, 25-39. 1983.

[8] Daubechies, I. *Ten lectures on wavelets.* Society for Industry and Applied Math, Philadephia, PA 1992.

[9] Daugman, J. "Uncertainty relation for resolution in space, spatial frequency, and orientation optimized by two-dimensional visual cortical filters. *Journal of Optical Soc. Amer.* Vol.2, No.7, 1985.

[10] Diaconis, P and Freedman, D. "On the Statistics of Vision: the Julesz Conjecture", *Journal of Math. Psychology* vol. 24. 1981.

[11] Donoho, D.L. and Johnstone, I.M. "Ideal de-noising in an orthonormal basis chosen from a libary of bases. " *Acad. Sci. Paris, Ser I*, Vol.319, pp1317-1322,1994.

[12] Field, D. "Relations between the statistics of natural images and the response properties of cortical cells", *J of Opt. Soc. Amer.,* Vol.4, No.12, 1987.

[13] Gabor, D. "Theory of communication." IEE Proc.vol 93, no26, 1946.

[14] Geman, S. and Geman, D. "Stochastic relaxation, Gibbs distribution, and the Bayesian restoration of images." *IEEE Trans. PAMI,* 6, 721-741. 1984.

[15] Geman, S. and Graffigne, C. "Markov random field image models and their applications to computer vision." *Proc. Int. Congress of Math.*, Berkeley, CA, 1986.

[16] Geyer, C. J. and Thompson, E. A. "Annealing Markov chain Monto Carlo with applications to ancestral inference." *J. of Amer. Stat. Assoc.*, **90**, 909-920. 1995.

[17] Haralick, R. M. "Statistics and structural approach to texture." *Proc. IEEE*, **67**, 786-804. 1979.

[18] Heeger, D. J. and Bergen, J. R. "Pyramid-based texture analysis/synthesis." *Computer Graphics*, in press. 1995.

[19] Jain, A. K. and Farrokhsia, F. "Unsupervised texture segmentation using Gabor filters." *Pattern Recognition*, **24**, 1167-1186. 1991

[20] Jaynes, E.T. "Information theory and statistical mechanics". *Physical Review.* **106**, pp620-630, 1957.

[21] Julesz, B. "Visual pattern discrimination." *IRE Trans. of Information Theory* IT-8, pp84-92, 1962.

[22] Kullback, S and Leibler, R. A. "On information and sufficiency", *Annual Math. Stat.* vol.22, pp79-86, 1951.

[23] Lee, T. S. Image representation using 2D Gabor wavelets. To appear in *IEEE Trans. of Pattern Analysis and Machine Intelligence.*

[24] Mallat, S. "Mumtiresolution approximations and wavelet orthonormal bases of $L^2(R)$." *Trans. Amer. Math. Soc.* **315**, 69-87. 1989.

[25] Mao, J. and Jain, A. K. Texture classification and segmentation using multiresolution simultaneous autoregressive models. *Pattern recognition*, **25**, 173-188. 1992.

[26] McCormick B. H. and Jayaramamurthy, S. N. "Time series models for texture synthesis." *Int. J. Comput. Inform. Sci.* **3**, 329-343. 1974.

[27] K. Popat and R.W. Picard, "Novel cluster-based probability model for texture synthesis, classification, and compression." Proc. SPIE Visual Comm., Cambridge, MA. 1993.

[28] Qian, W. and Titterington, D. M. "Multidimensional Markov chain models for image textures." *J. Royal Stat. Soc., B*, **53**, 661-674. 1991.

[29] Silverman, M. S., Grosof, D. H., De Valois, R. L., and Elfar, S. D. "Spatial-frequency organization in primate striate cortex." *Proc. Natl. Acad. Sci. U.S.A.*, **86**. 1989.

[30] Simoncelli, E. P. Freeman,W.T, Adelson, E. H, Heeger, D.J. "Shiftable multiscale transforms. *IEEE Trans. on information theory.* Vol. 38, pp587-607, 1992.

[31] Tsatsanis, M. K. and Giannakis, G. B. "Object and texture classification using higher order statistics." *IEEE Trans on PAMI*, **7**, 733-749. 1992.

[32] Winkler, G. *Image Analysis, Random Fields and dynamic Monte Carlo Methods*, Springer-Verlag 1995.

[33] Witkin, A. and Kass, M. "Reaction-diffusion textures." *Computer Graphics*, **25**, 299-308. 1991.

[34] Yuan, J. and Rao S. T. "Spectral estimation for random fields with applications to Markov modeling and texture Classification." *Markov Random Fields*, Chellappa and Jain ed. 179-209. 1993.

[35] Zhu, S. C. and Yuille, A.L "Region Competition: unifying snakes, region growing, and Bayes/MDL for multi-band image segmentation". *IEEE Trans. on PAMI.* Vol.18, No.9, 1996.

[36] Zhu, S.C., Wu, Y.N. and Mumford,D.B. "Minimax Entropy Principle and its applications", *Harvard Robotics Lab. Technique Report*, 1996.