

Exploring Texture Ensembles by Efficient Markov Chain Monte Carlo—Toward a “Trichromacy” Theory of Texture

Song Chun Zhu, Xiu Wen Liu, and Ying Nian Wu

Abstract—This article presents a mathematical definition of texture—the *Julesz ensemble* $\Omega(\mathbf{h})$, which is the set of all images (defined on Z^2) that share identical statistics \mathbf{h} . Then texture modeling is posed as an inverse problem: Given a set of images sampled from an unknown Julesz ensemble $\Omega(\mathbf{h}_*)$, we search for the statistics \mathbf{h}_* which define the ensemble. A Julesz ensemble $\Omega(\mathbf{h})$ has an associated probability distribution $q(\mathbf{I}; \mathbf{h})$, which is uniform over the images in the ensemble and has zero probability outside. In a companion paper [33], $q(\mathbf{I}; \mathbf{h})$ is shown to be the *limit distribution* of the FRAME (Filter, Random Field, And Minimax Entropy) model [36], as the image lattice $\Lambda \rightarrow Z^2$. This conclusion establishes the intrinsic link between the scientific definition of texture on Z^2 and the mathematical models of texture on finite lattices. It brings two advantages to computer vision: 1) The engineering practice of synthesizing texture images by matching statistics has been put on a mathematical foundation. 2) We are released from the burden of learning the expensive FRAME model in feature pursuit, model selection and texture synthesis. In this paper, an efficient Markov chain Monte Carlo algorithm is proposed for sampling Julesz ensembles. The algorithm generates random texture images by moving along the directions of filter coefficients and, thus, extends the traditional single site Gibbs sampler. We also compare four popular statistical measures in the literature, namely, moments, rectified functions, marginal histograms, and joint histograms of linear filter responses in terms of their descriptive abilities. Our experiments suggest that a small number of bins in marginal histograms are sufficient for capturing a variety of texture patterns. We illustrate our theory and algorithm by successfully synthesizing a number of natural textures.

Index Terms—Gibbs ensemble, Julesz ensemble, texture modeling, texture synthesis, Markov chain Monte Carlo.

1 INTRODUCTION AND MOTIVATIONS

IN his seminal paper of 1962 [19], Julesz initiated research on texture by asking the following fundamental question:

“What features and statistics are characteristic of a texture pattern, so that texture pairs that share the same features and statistics cannot be told apart by preattentive human visual perception?”¹

Julesz’s question has been the scientific theme in texture modeling and perception in the last three decades. His question raised two major challenges. The first is in psychology and neurobiology: What *features and statistics* are the *basic elements* in human texture perception? The second lies in mathematics and statistics: Given a set of consistent statistics, how do we generate random texture images with identical statistics? In mathematical language, we should be able to explore the *ensemble of texture images* that

1. Preattentive vision is referred to some psychophysics phenomena that early stage visual processing seems to be accomplished simultaneously (for the entire image independent of the number of stimuli) and automatically (without attention being focused on any one part of the image). See [30] for a discussion.

- S.C. Zhu and X.W. Liu are with the Department of Computer and Information Sciences, The Ohio State University, Columbus, OH 43210. E-mail: {szhu, liux}@cis.ohio-state.edu
- Y.N. Wu is with the Department of Statistics, University of California, Los Angeles, CA 90095. E-mail: ywu@math.ucla.edu.

Manuscript received 19 Mar. 1999; accepted 24 Feb. 2000.

Recommended for acceptance by R. Picard.

For information on obtaining reprints of this article, please send e-mail to: tpami@computer.org, and reference IEEECS Log Number 109435.

have exactly the same statistics. The first question cannot be answered without solving the second one. As Julesz conjectured [21], the ideal theory of texture should be similar to the theory of trichromacy, which states that any visible color is a linear combination of three basic colors: red, green, and blue. In texture theory, this corresponds to the search for 1) the basic texture statistics and 2) a method for mixing the exact amount of statistics in a given recipe.

The search for features and statistics has gone a long way beyond Julesz 2-gon statistics conjecture. Examples include co-occurrence matrices, run-length statistics [32], sizes and orientations of various textons [31], cliques in Markov random fields [8], as well as dozens of other measures. All these features have rather limited expressive power. A quantum jump occurred in the 1980s when Gabor filters [11], filter pyramids, and wavelet transforms [10] were introduced to image representation. Another advance occurred in the 1990s when simple statistics, e.g., first and second order moments, were replaced by histograms (either marginal or joint) [18], [35], [9], which contain all the higher order moments.

Research on mathematical methods for rendering texture pairs with identical statistics has also been extensive in the literature. The earliest work was task specific. For example, the “4-disk method” was designed for rendering random texture pairs sharing 2-gon statistics [5]. In computer vision, many systematic methods have been developed. For example:

1. Gagalowicz and Ma [12] used a steepest descent minimization of the sum of squared errors of intensity histograms and two-pixel auto-correlation matrices.

2. Heeger and Bergen [18] used pyramid collapsing to match marginal histograms of filter responses.
3. Anderson and Langer [2] used steepest descent method to minimize match errors of rectified functions.
4. De Bonet and Viola [9] matched full joint histogram using Markov trees.
5. Portilla and Simoncelli [29] matched various correlations through repeated projections onto statistics constrained surfaces.

Despite the successes of these methods in texture synthesis and their computational convenience, the following three fundamental questions remain unanswered:

1. What is the mathematical definition of texture adopted in the above methods? Is it technically sound to synthesize textures by minimizing statistics errors, without explicit statistical modeling?
2. How are the above texture synthesis methods related to the rigorous mathematical models of textures, for example, Markov random field models [8] and minimax entropy models [36]?
3. In practice, the above methods for synthesizing textures do not guarantee a close match of statistics, nor do they intend to **sample** the ensemble of images with identical statistics. Is there a general way of designing algorithms for efficient statistics matching and image sampling?

This paper answers problems 1 and 3 and briefly reviews the answer to question 2. A detailed study of question 2 is referred to a companion paper [33].

First, we define the Julesz ensemble. Given a set of statistics \mathbf{h} extracted from a set of observed images of a texture pattern, such as histograms of filter responses, a Julesz ensemble is defined as the set of all images (defined on Z^2) that share the same statistics as the observed. A Julesz ensemble, denoted by $\Omega(\mathbf{h})$, has an associated probability distribution $q(\mathbf{I}; \mathbf{h})$ which is uniform over the images in the ensemble and has zero probability outside. The Julesz ensemble leads to a mathematical definition of texture. In a companion paper [33], we show that the Julesz ensemble (or the uniform distribution $q(\mathbf{I}; \mathbf{h})$) is equivalent to the Gibbs ensemble, and the latter is the *limit* of a FRAME (Filter, Random Field, And Minimax Entropy) model $p(\mathbf{I}; \beta)$ as the image lattice goes to infinity. The ensemble equivalence reveals two significant facts in texture modeling and synthesis.

1. On large image lattices, we can draw images from the Julesz ensemble $\Omega(\mathbf{h})$ without learning the expensive FRAME model. These sampled images are typical of the corresponding Gibbs model $p(\mathbf{I}; \beta)$, and can be used for texture synthesis, feature pursuit, and model selection [36].
2. For a large (say, 256×256 pixels) image sampled from the Julesz ensemble, any local patch of the image given its environment follows the FRAME model derived by the minimax entropy principle. Thus, the FRAME model is an inevitable model for textures on finite lattices.

Second, this paper proposes an efficient method for sampling from the Julesz ensemble by Markov chain Monte

Carlo (MCMC). In the traditional single-site Gibbs sampler [13], [36], the MCMC transition is designed in the following way: At each step, a pixel is chosen at random or in a fixed scan order, and the intensity value of the pixel is updated according to its conditional distribution given the intensities of neighboring pixels. Since the filters used in the texture models are often very large (e.g., 32×32 pixels), flipping one pixel at a time has very little effect on the filter responses. As a result, the single site Gibbs sampler can be very inefficient, and the formation and change of local texture features may take a long time.

Motivated by the recent work of Liu and Wu [24] and Liu and Sabatti [25] (see also, the references therein), we approach the efficiency problem by designing conditional moves along the directions of filter coefficients. For a linear filter, the window function² spans one dimension in the image space. Thus, for a set of K filters, we have K axes at each pixel, and these axes do not have to be orthogonal to each other. We propose random moves along these axes, and thus, the proposed moves update large patches of the image so that local features can be formed and changed quickly. Third, we compare four popular statistical measures in the literature: moments, rectified functions, marginal histograms, and joint histograms of Gabor filter responses. Our experiments show that moments and rectified functions are not sufficient for texture modeling. We shall also pursue the minimum set of statistics that can describe texture patterns; our experiments demonstrate that a small number of bins in the marginal histograms are sufficient for capturing a variety texture patterns, so the full joint histogram appears to be an over-fit. We demonstrate our theory and algorithm by synthesizing many natural textures.

The paper is organized as follows: We start with a discussion of features and statistics in Section 2 which is followed by a mathematical study of texture modeling in Section 3. Section 4 shows a group of experiments on texture synthesis using the Gibbs sampler. Section 5 describes feature pursuit experiments in selecting the histogram bins and compares a variety of statistics. Section 6 presents a generalized Gibbs sampler for efficient MCMC sampling. Finally, we conclude the paper with a discussion in Section 7.

2 IMAGE FEATURES AND STATISTICS

To pursue a “trichromacy” theory for texture, in this section we review some important image features and statistics that have been used in texture modeling.

Let \mathbf{I} be an image defined on a finite lattice $\Lambda \subset Z^2$. For each pixel $v = (x, y) \in \Lambda$, the intensity value at v is denoted by $\mathbf{I}(v) \in S$, with S being a finite interval on the real line or a finite set of quantized grey levels. We denote by $\Omega_\Lambda = S^{|\Lambda|}$ the space of all images on Λ .

In modeling homogeneous texture images, we start with exploring a finite set of statistics of some local image features. Fig. 1 shows three major categories of image features studied in the literature.

The first category consists of k -gons proposed by Julesz. A k -gon is a polygon of k vertices indexed by $\alpha = (u_1, u_2, \dots, u_k)$, where $u_i = (\Delta x_i, \Delta y_i)$ is the displacement of the i th vertex if

2. This is called the impulse response in engineering and the receptive field in neurosciences.

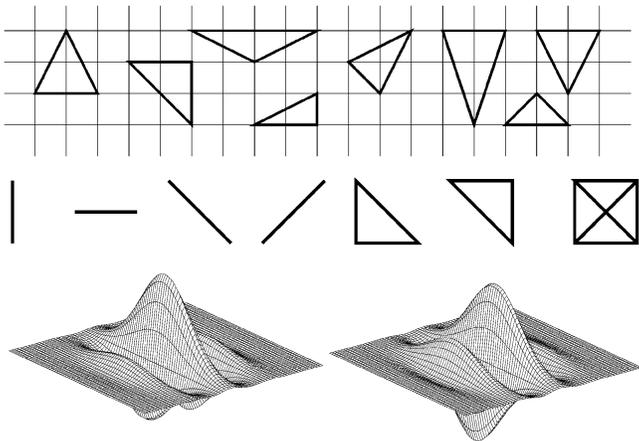


Fig. 1. Choices of image features. Top row: various 3-polygons on the image lattice. Middle row: cliques in Markov random fields. Bottom row: Gabor filters with cosine (left) and sine (right) components.

we put the center of the k -gon at the origin. Δx_i and Δy_i must be integers. If we move this k -gon on the lattice, under some boundary conditions we collect a set of k -tuples

$$\{(\mathbf{I}(v + u_1), \mathbf{I}(v + u_2), \dots, \mathbf{I}(v + u_k)); v \in \Lambda\}.$$

The k -gon statistic is the k -dimensional joint intensity histogram of these k -tuples and it is also called the *co-occurrence matrix*. To be more specific, if we quantize the intensity value into $1, 2, \dots, g$, then the k -gon statistic of \mathbf{I} is expressed as

$$\mathbf{h}^{(\alpha)}(b_1, b_2, \dots, b_k; \mathbf{I}) = \frac{1}{|\Lambda|} \sum_{v \in \Lambda} \prod_{i=1}^k \delta(b_i - \mathbf{I}^{(i)}(v + u_i)),$$

where $\delta()$ is the Dirac delta function with unit mass at zero and zero elsewhere. We assume that boundary conditions are properly handled (e.g., periodic boundary condition). Fig. 1 displays a set of triangles in the top row. One co-occurrence matrix is computed for each type of polygon. The k -gon statistics suffer from the curse of dimensionality for even a small k . For instance, for k as small as 4, and $g = 10$, the dimensionality of a 4-gon statistic is comparable to the size of the image. Summarizing the image into k -gon statistics can hardly achieve any data reduction.

The second type of features are the cliques in Markov random fields (MRF), as shown in the middle row of Fig. 1. Given a neighborhood system on the lattice Λ , a clique is a set of pixels that are neighbors of each other, so a clique is a special type of k -gon. Let $\alpha = (u_1, u_2, \dots, u_{k_\alpha})$ be the index for different types of cliques under a neighborhood system. According to the Hammersley-Clifford theorem [3], a Markov random field model has the Gibbs form

$$p(\mathbf{I}) = \frac{1}{Z} e^{-\sum_{\alpha} \sum_{v \in \Lambda} U_{\alpha}(\mathbf{I}(v+u_1), \dots, \mathbf{I}(v+u_{k_{\alpha}}))},$$

where Z is the normalization constant or the partition function and U_{α} are potential functions of k_{α} variables. The above Gibbs distribution can be derived from the maximum entropy principle under the constraints that $p(\mathbf{I})$ reproduces, on average, the co-occurrence matrices $\mathbf{h}^{(\alpha)}(b_1, \dots, b_{k_{\alpha}}; \mathbf{I}), \forall \alpha$. Therefore, the Gibbs model integrates all

the co-occurrence matrices for the cliques into a single probability distribution. See Picard et al. [27] for a related result. Like k -gon statistics, this general MRF model also suffers from curse of dimensionality even for small cliques. The existing MRF texture models are much simplified in order to reduce the dimensionality of potential functions, such as in autobinomial models [8], Gaussian MRF models [6] and ϕ -models [14].

The co-occurrence matrices (or joint intensity histograms) on the polygons and cliques have been proven inadequate for describing real world images and irrelevant to biologic vision systems. In the late 1980s, it was realized that real world imagery is better represented by spatial/frequency bases, such as Gabor filters [11], wavelet transforms [10], and filter pyramids. These filters are often called *image features*. Given a set of filters $\{F^{(\alpha)}, \alpha = 1, 2, \dots, K\}$, a subband image $\mathbf{I}^{(\alpha)} = F^{(\alpha)} * \mathbf{I}$ is computed for each filter $F^{(\alpha)}$.

Thus, the third method in texture analysis extracts statistics on subband images or pyramid instead of the intensity image. From a dimension reduction perspective, the filters characterize local texture features, as a result, very simple statistics of the subband images can capture information that would otherwise require k -gon or clique statistics of very high dimensions.

While Gabor filters are well-grounded in biological vision [7], very little is known about how visual cortices pool statistics across images. Fig. 2 displays four popular choices of statistics in the literature.

1. Moments of a single filter response, e.g., mean and variance of $\mathbf{I}^{(\alpha)}$ in Fig. 2a,

$$\mathbf{h}^{(\alpha,1)}(\mathbf{I}) = \frac{1}{|\Lambda|} \sum_{v \in \Lambda} \mathbf{I}^{(\alpha)}(v),$$

$$\mathbf{h}^{(\alpha,2)}(\mathbf{I}) = \frac{1}{|\Lambda|} \sum_{v \in \Lambda} (\mathbf{I}^{(\alpha)}(v) - \mathbf{h}^{(\alpha,1)})^2.$$

2. Rectified functions that resemble responses of "on/off" cells [2]:

$$\mathbf{h}^{(\alpha,+)}(\mathbf{I}) = \frac{1}{|\Lambda|} \sum_{v \in \Lambda} R^+(\mathbf{I}^{(\alpha)}(v)),$$

$$\mathbf{h}^{(\alpha,-)}(\mathbf{I}) = \frac{1}{|\Lambda|} \sum_{v \in \Lambda} R^-(\mathbf{I}^{(\alpha)}(v)),$$

where the functions $R^+()$, $R^-()$ are shown in Fig. 2b.

3. One bin of the empirical histogram of $\mathbf{I}^{(\alpha)}$,

$$\mathbf{h}^{(\alpha)}(b; \mathbf{I}) = \frac{1}{|\Lambda|} \sum_{v \in \Lambda} \delta(b - \mathbf{I}^{(\alpha)}), \quad \forall \alpha,$$

where $\delta()$ is a window function shown in Fig. 2c.

4. One bin of the full joint histogram,

$$\mathbf{h}(b_1, b_2, \dots, b_k; \mathbf{I}) = \frac{1}{|\Lambda|} \sum_{v \in \Lambda} \prod_{i=1}^k \delta(b_i - \mathbf{I}^{(i)}(v)), \quad (1)$$

where (b_1, b_2, \dots, b_k) is the index for one bin in a k -dimensional histogram in Fig 2d.

Perhaps the most general statistics are the co-occurrence matrices (histograms) for polygons whose vertices are on the image pyramid across multiple layers, as displayed in

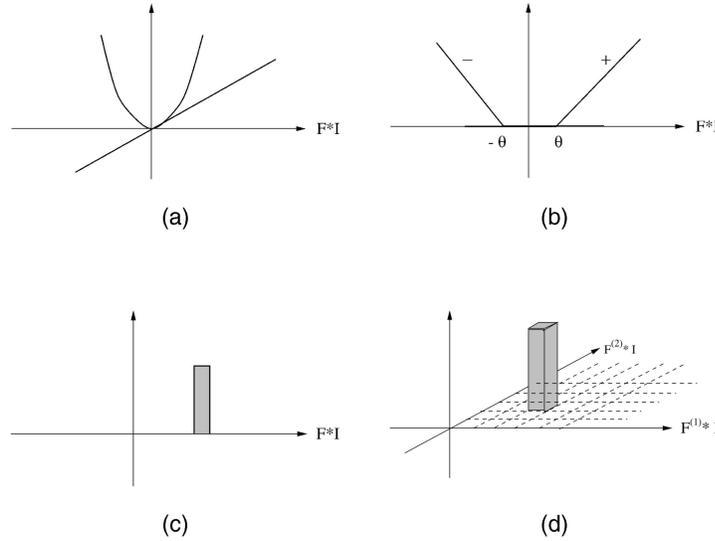


Fig. 2. Four choices of statistics in the feature space: (a) Moments, (b) rectified functions for the “on/off” cells, (c) a bin of the marginal histogram using rectangle window or Gaussian window, and d) a bin of joint histogram. This figure is generalized from [2].

Fig. 3. For a k -gon in the image pyramid, a cooccurrence matrix (or joint histogram) can be computed,

$$\mathbf{h}^{(\alpha,k)}(b_1, b_2, \dots, b_k; \mathbf{I}) = \frac{1}{|\Lambda|} \sum_{v \in \Lambda} \prod_{i=1}^k \delta(b_i - \mathbf{I}^{(l_i)}(v + u_i)). \quad (2)$$

In the above definition, $\alpha = ((l_1, u_1), (l_2, u_2), \dots, (l_k, u_k))$ is the index for the polygons in the pyramid, where l_i, u_i are respectively the indexes for the subband and the displacement of the i th vertex of the polygon. It is easy to see all the traditional co-occurrence matrices and histograms are special cases of the statistics in (2). For example, it reduces to the traditional co-occurrence matrix when the pyramid contains only the intensity image \mathbf{I} ; it reduces to the full joint histogram [9] in (1) when the polygon α is a straight line crossing the pyramid; it captures spatial correlations [29], such as parallelism, if the polygon has two straight lines crossing the pyramid as illustrated by the rightmost polygon in Fig. 3.

Obviously, joint statistics and correlations are useful in aligning image features crossing spatial and frequency domains, such as edges and some elaborate details of texture elements. The question is how far one should

pursue these high order statistics without sophisticated dimension reduction (such as textons). The complexity of the statistics are limited by both the computational complexity and the statistical efficiency in estimating the model with finite data.

In the literature, Heeger and Bergen made the first attempt to generate textures using marginal histograms [18] and Zhu et al. studied a new class of Markov random field model that can reproduce marginal statistics [36].³ Recently, De Bonet and Viola [9] and Simoncelli and Portilla [29] have argued that joint statistics and correlations are needed for synthesizing some texture patterns. The argument is mainly based on the fact that matching marginal statistics for the filters of the steerable pyramid used by Heeger and Bergen [18] cannot reproduce some texture patterns. This argument is questionable because computationally, the Heeger and Bergen algorithm does not guarantee a close match of statistics, nor is it intended to *sample* from a texture ensemble. In theory, it is provable that marginal statistics are sufficient for reconstructing the full probability distribution of a texture [36]. Of course, this conclusion does not necessarily prevent us from using joint statistics or correlations between filter responses. Then there are two key questions that need to be studied: 1) What is the minimum set of statistics for defining a texture pattern? 2) How can we sample textures unbiasedly from the set of texture images that share identical statistics? Essentially, we need to make sure that we are not sampling from a special subset.

3 JULESZ ENSEMBLE AND MCMC SAMPLING OF TEXTURE

As the second step to pursue the “trichromacy” theory of texture, in this section, we first propose a mathematical definition of texture—the Julesz ensemble, and then we

3. In FRAME [36], it is straightforward to derive Markov random field models that match other statistics discussed in this section.

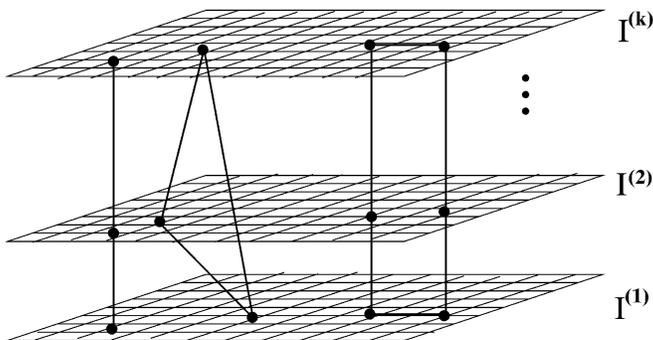


Fig. 3. The most general statistics are co-occurrence matrices for polygons in an image pyramid.

study an algorithm for sampling images from the Julesz ensemble.

3.1 Julesz Ensemble—A Mathematical Definition of Texture

Given a set of K statistics $\mathbf{h} = \{\mathbf{h}^{(\alpha)} : \alpha = 1, 2, \dots, K\}$ which have been normalized with respect to the size of the lattice $|\Lambda|$, an image \mathbf{I} is mapped into a point $\mathbf{h}(\mathbf{I}) = (\mathbf{h}^{(1)}(\mathbf{I}), \dots, \mathbf{h}^{(K)}(\mathbf{I}))$ in the *space of statistics*. Let

$$\Omega_{\Lambda}(\mathbf{h}_0) = \{\mathbf{I} : \mathbf{h}(\mathbf{I}) = \mathbf{h}_0\}$$

be the set of images sharing the same statistics \mathbf{h}_0 . Then, the image space Ω_{Λ} is partitioned into equivalence classes

$$\Omega_{\Lambda} = \cup_{\mathbf{h}} \Omega_{\Lambda}(\mathbf{h}).$$

Due to intensity quantization in finite lattices, we relax the constraint on statistics and define the image set as

$$\Omega_{\Lambda}(\mathcal{H}) = \{\mathbf{I} : \mathbf{h}(\mathbf{I}) \in \mathcal{H}\},$$

where \mathcal{H} is an open set around \mathbf{h}_0 .

$\Omega_{\Lambda}(\mathcal{H})$ implies a uniform distribution

$$q(\mathbf{I}; \mathcal{H}) = \begin{cases} \frac{1}{|\Omega_{\Lambda}(\mathcal{H})|} & \text{for } \mathbf{I} \in \Omega_{\Lambda}(\mathcal{H}), \\ 0 & \text{otherwise} \end{cases}$$

where $|\Omega_{\Lambda}(\mathcal{H})|$ is the volume of the set.

Definition. Given a set of normalized statistics $\mathbf{h} = \{\mathbf{h}^{(\alpha)} : \alpha = 1, 2, \dots, K\}$, a Julesz ensemble $\Omega(\mathbf{h})$ is the limit of $\Omega_{\Lambda}(\mathcal{H})$ as $\Lambda \rightarrow \mathbb{Z}^2$ and $\mathcal{H} \rightarrow \{\mathbf{h}\}$ under some boundary conditions.

A Julesz ensemble $\Omega(\mathbf{h})$ is a mathematical idealization of $\Omega_{\Lambda}(\mathcal{H})$ on a large lattice with \mathcal{H} close to \mathbf{h} . As $\Lambda \rightarrow \mathbb{Z}^2$, it makes sense to let the normalized statistics $\mathcal{H} \rightarrow \{\mathbf{h}\}$. We assume $\Lambda \rightarrow \mathbb{Z}^2$ in the sense of van Hove [15], i.e., the ratio between the size of the boundary and the size of Λ goes to 0, $|\partial\Lambda|/|\Lambda| \rightarrow 0$. In engineering practice, we often consider a lattice big enough if $\frac{|\partial\Lambda|}{|\Lambda|}$ is very small, e.g., 1/15. Thus, with a slight abuse of notation and also to avoid technicalities in dealing with limits, we consider a sufficiently large image (e.g., 256×256 pixels) as an infinite image in the rest of the paper. See the companion paper [33] for a more careful treatment.

A Julesz ensemble $\Omega(\mathbf{h})$ defines a texture pattern on \mathbb{Z}^2 and it maps textures into the space of feature statistics \mathbf{h} . By analogy to color, as an electromagnetic wave with wavelength $\lambda \in [400, 700]$ nm defines a unique visible color, a statistic value \mathbf{h} defines a texture pattern⁴. We shall study the relation between the Julesz ensemble and the mathematical models of texture in the next section.

A mathematical definition of texture could be different from a texture category in human texture perception. The latter has the coarser precision on the statistics \mathbf{h} and is often influenced by experience. For example, Julesz proposed that texture pairs which are not preattentively segmentable belong to the same category. Recently, many groups have reported that texture pairs which are not preattentively segmentable by naive subjects become

4. We named this ensemble after Julesz to remember his pioneering work on texture. This does not necessarily mean that Julesz defined texture pattern with this mathematical formulation.

segmentable after practice [22]. This phenomenon is similar to color perception.

With the mathematical definition of texture, texture modeling is posed as an inverse problem. Suppose we are given a set of observed training images

$$\Omega_{\text{obs}} = \{\mathbf{I}_{\text{obs},1}, \mathbf{I}_{\text{obs},2}, \dots, \mathbf{I}_{\text{obs},M}\},$$

which are sampled from an unknown Julesz ensemble $\Omega_* = \Omega(\mathbf{h}_*)$. The objective of texture modeling is to search for the statistics \mathbf{h}_* .

We first choose a set of K statistics from a dictionary B discussed in Section 2. We then compute the normalized statistics over the observed images $\mathbf{h}_{\text{obs}} = (\mathbf{h}_{\text{obs}}^{(1)}, \dots, \mathbf{h}_{\text{obs}}^{(K)})$, with

$$\mathbf{h}_{\text{obs}}^{(\alpha)} = \frac{1}{M} \sum_{i=1}^M \mathbf{h}^{(\alpha)}(\mathbf{I}_{\text{obs},i}), \quad \alpha = 1, 2, \dots, K. \quad (3)$$

Then, we define an ensemble of texture images using \mathbf{h}_{obs} ,

$$\Omega_{K,\epsilon} = \{\mathbf{I} : D(\mathbf{h}^{(\alpha)}(\mathbf{I}), \mathbf{h}_{\text{obs}}^{(\alpha)}) \leq \epsilon, \quad \forall \alpha\}, \quad (4)$$

where D is some distance, such as the L_1 distance for histograms. If Λ is large enough to be considered infinite, we can set ϵ essentially at 0, and we denote the corresponding $\Omega_{K,\epsilon}$ as Ω_K . The ensemble Ω_K implies a uniform probability distribution $q(\mathbf{I}; \mathbf{h})$ over Ω_K , whose entropy is $\log |\Omega_K|$.

To search for the underlying Julesz ensemble Ω_* , one can adopt a pursuit strategy used by Zhu et al. [36]. When $k = 0$, we have $\Omega_0 = \Omega_{\Lambda}$. Suppose at step k , a statistic \mathbf{h} is chosen, then at step $k+1$ a statistic $\mathbf{h}^{(k+1)}$ is added to have $\mathbf{h}_+ = (\mathbf{h}, \mathbf{h}^{(k+1)})$. $\mathbf{h}^{(k+1)}$ is selected for the largest entropy decrease among all statistics in the dictionary B ,

$$\begin{aligned} \mathbf{h}^{(k+1)} &= \arg \max_{\beta \in B} [\text{entropy}(q(\mathbf{I}; \mathbf{h})) - \text{entropy}(q(\mathbf{I}; \mathbf{h}_+))] \\ &= \arg \max_{\beta \in B} [\log |\Omega_k| - \log |\Omega_{k+1}|]. \end{aligned} \quad (5)$$

The decrease of entropy is called the *information gain* of $\mathbf{h}^{(k+1)}$.

As shown in Fig. 4, as more statistics are added, the entropy or volume of the Julesz ensemble decreases monotonically

$$\Omega_{\Lambda} = \Omega_0 \supseteq \Omega_1 \supseteq \dots \supseteq \Omega_k \supseteq \dots$$

Obviously, introducing too many statistics will lead to an “over-fit.” In the limit of $k \rightarrow \infty$, Ω_{∞} only includes the observed images in Ω_{obs} and their translated versions.

With the observed finite images, the choice of statistics \mathbf{h} and the Julesz ensemble $\Omega(\mathbf{h})$ is an issue of model complexity that has been extensively studied in the statistics literature. In the minimax entropy model [36], [35], an AIC criterion [1] is adopted for model selection. The intuitive idea of AIC is simple. With finite images, we should measure the fluctuation of the new statistics $\mathbf{h}^{(k+1)}$ over the training images in Ω_{obs} . Thus when a new statistic is added, it brings information as well as estimation error. The feature pursuit process should stop when the estimation error brought by $\mathbf{h}^{(k+1)}$ is larger than its information gain.

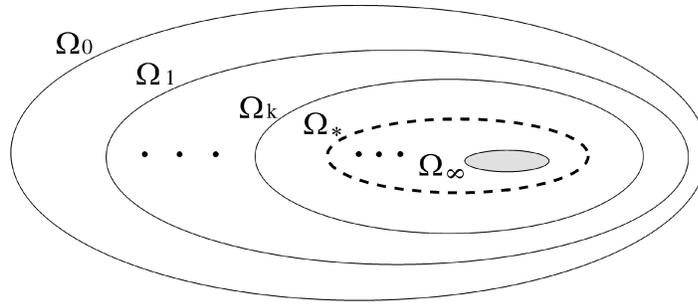


Fig. 4. The volume (or entropy) of Julesz ensemble decreases monotonically with more statistical constraints added.

3.2 The Gibbs Ensemble and Ensemble Equivalence

To make this paper self-contained, we briefly discuss in this section the Gibbs ensemble and the equivalence between the Julesz and Gibbs ensembles. A detailed study is referred to a companion paper [33].

Given a set of observed images Ω_{obs} and the statistics \mathbf{h}_{obs} , another line of research is to pursue probabilistic texture models, in particular the Gibbs distributions or Markov Random Field (MRF) models.

One general class of MRF model is the FRAME model studied by Zhu et al. [35], [36]. The FRAME model derived from the maximum entropy principle has the Gibbs form

$$\begin{aligned} p(\mathbf{I}; \beta) &= \frac{1}{Z(\beta)} \exp\left\{-\sum_{\alpha=1}^K \langle \beta^{(\alpha)}, \mathbf{h}^{(\alpha)}(\mathbf{I}) \rangle\right\} \\ &= \frac{1}{Z(\beta)} \exp\{\langle \beta, \mathbf{h}(\mathbf{I}) \rangle\}. \end{aligned} \quad (6)$$

The parameters $\beta = (\beta^{(1)}, \beta^{(2)}, \dots, \beta^{(K)})$ are Lagrange multipliers. The values of β are determined so that $p(\mathbf{I}; \beta)$ reproduces the observed statistics,

$$E_{p(\mathbf{I}; \beta)}[\mathbf{h}^{(\alpha)}(\mathbf{I})] = \mathbf{h}_{\text{obs}}^{(\alpha)} \quad \alpha = 1, 2, \dots, K. \quad (7)$$

The selection of statistics is guided by a minimum entropy principle.

As the image lattice becomes large enough, the fluctuations of the normalized statistics diminish. Thus as $\Lambda \rightarrow \mathbb{Z}^2$, the FRAME model converges to a *limiting random field* in the absence of phase transition. The limiting random field essentially concentrates all its probability mass uniformly over a set of images which we call the *Gibbs ensemble*.⁵ In a companion paper [33], we proved that the Gibbs ensemble given by $p(\mathbf{I}; \beta)$ is equivalent to the Julesz ensemble specified by $q(\mathbf{I}; \mathbf{h}_{\text{obs}})$. The relationship between β and \mathbf{h}_{obs} is expressed in (7). Intuitively, $q(\mathbf{I}; \mathbf{h}_{\text{obs}})$ is defined by a “hard” constraint, while the Gibbs model $p(\mathbf{I}; \beta)$ is defined by a “soft” constraint. Both use the observed statistics \mathbf{h}_{obs} , and the model $p(\mathbf{I}; \beta)$ concentrates on the Julesz ensemble uniformly as the lattice Λ gets big enough.

5. In the computation of a feature statistic $\mathbf{h}(\mathbf{I})$, we need to define boundary conditions so that the filter responses in Λ are well defined. In case of phase transition, the limit of a Gibbs distribution is not unique, and it depends on the boundary conditions. However, the equivalence between Julesz ensemble and Gibbs ensemble holds even with phase transition. The study of phase transition is beyond the scope of this paper.

The ensemble equivalence reveals two significant facts in texture modeling:

1. Given a set of statistics \mathbf{h} , we can synthesize typical texture images of the fitted FRAME model by sampling from the Julesz ensemble $\Omega(\mathbf{h})$ without learning the parameters β in the FRAME model [36]. Thus feature pursuit, model selection, and texture synthesis can be done effectively with the Julesz ensemble.
2. For images sampled from a Julesz ensemble, a local patch of the image given its environment follows the Gibbs distribution (or FRAME model) derived by the minimax entropy principle. Therefore, the Gibbs model $p(\mathbf{I}; \beta)$ provides a parametric form for the conditional distribution of $q(\mathbf{I}; \mathbf{h})$ on small image patches. $p(\mathbf{I}; \beta)$ should be used for tasks such as texture classification and segmentation.

The pursuit of Julesz ensembles can also be based on the minimax entropy principle. First, the definition of $\Omega(\mathbf{h})$ as the *maximum* set of images sharing statistics \mathbf{h} is equivalent to a maximum entropy principle. Second, the pursuit of statistics in (5) uses a minimum entropy principle. Therefore, a unifying picture emerges for texture modeling under the minimax entropy theory.

3.3 Sampling the Julesz Ensemble

Sampling the Julesz ensemble is by no means a trivial task! As $|\Omega_K|/|\Omega_\Lambda|$ is exponentially small, the Julesz ensemble has almost zero volume in the image space. Thus rejection sampling methods are inappropriate and we resort to Markov chain Monte Carlo methods.

First, we define a function

$$G(\mathbf{I}) = \begin{cases} 0, & \text{if } D(\mathbf{h}^{(\alpha)}(\mathbf{I}), \mathbf{h}_{\text{obs}}^{(\alpha)}) \leq \epsilon, \forall \alpha \\ \sum_{\alpha=1}^K D(\mathbf{h}^{(\alpha)}(\mathbf{I}), \mathbf{h}_{\text{obs}}^{(\alpha)}), & \text{otherwise.} \end{cases}$$

Then the distribution

$$q(\mathbf{I}; \mathbf{h}, T) = \frac{1}{Z(T)} \exp\{-G(\mathbf{I})/T\} \quad (8)$$

goes to a Julesz ensemble Ω_K , as the temperature T goes to 0. The $q(\mathbf{I}; \mathbf{h}, T)$ can be sampled by the Gibbs sampler or other MCMC algorithms.

Algorithm I: Sampling the Julesz Ensemble

Given texture images $\{\mathbf{I}_{\text{obs}, i}, i = 1, 2, \dots, M\}$.
Given K statistics (filters) $\{F^{(1)}, F^{(2)}, \dots, F^{(K)}\}$.

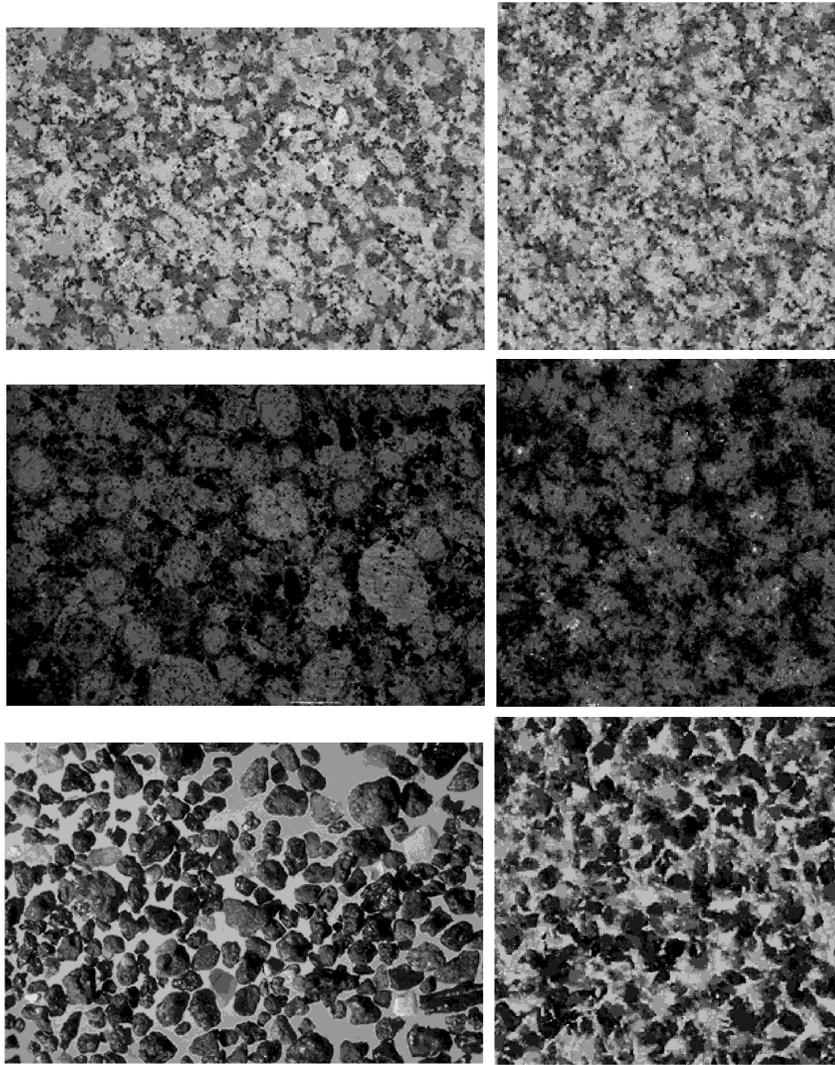


Fig. 5. Left column: the observed texture images. Right column: the synthesized texture images that share the exact histograms with the observed for 56 filters.

Compute $\mathbf{h}_{\text{obs}} = \{\mathbf{h}_{\text{obs}}^{(\alpha)}, \alpha = 1, \dots, K\}$.
 Initialize a synthesized image \mathbf{I} (e.g., white noise).
 $T \leftarrow T_0$
 Repeat
 Randomly pick a location $v \in \Lambda$,
 For $\mathbf{I}(v) \in S$ Do
 Calculate $q(\mathbf{I}(v) | \mathbf{I}(-v); \mathbf{h}, T)$.
 Randomly draw a new value of $\mathbf{I}(v)$ from
 $q(\mathbf{I}(v) | \mathbf{I}(-v); \mathbf{h}, T)$.
 Reduce T after each sweep.
 Record samples when $D(\mathbf{h}^{(\alpha)}(\mathbf{I}), \mathbf{h}_{\text{obs}}^{(\alpha)}) \leq \epsilon$ for
 $\alpha = 1, 2, \dots, K$.
 Until enough samples are collected.

In the above algorithm, $q(\mathbf{I}(v) | \mathbf{I}(-v); \mathbf{h}, T)$ is the conditional probability of the pixel value $\mathbf{I}(v)$ with intensities for the rest of the lattice fixed. A sweep flips $|\Lambda|$ pixels in a random visiting scheme or to flip all pixels in a fixed visiting scheme.

Due to the equivalence between the Julesz ensemble and the Gibbs ensemble [33], the sampled images from $q(\mathbf{I}; \mathbf{h})$ and those from $p(\mathbf{I}; \beta)$ share the same statistics in that they

produce not only the same statistics in \mathbf{h} , but also statistics extracted by any other filters, linear or nonlinear. It is worth emphasizing one key concept which has been misunderstood in some computer vision work: the Julesz ensemble is the set of “typical” images for the Gibbs model $p(\mathbf{I}; \beta)$, not the “most probable” images that minimize the Gibbs potential (or energy) in $p(\mathbf{I}; \beta)$.

One can use Algorithm I for selecting statistics \mathbf{h} , as in [36]. That is, one can pursue new statistics by decreasing the entropy as measured in (5). An in-depth discussion is referred to in [33].

4 EXPERIMENT: SAMPLING THE JULESZ ENSEMBLE

In our first set of experiments, we select all 56 linear filters (Gabor filters at various scales and orientations and small Laplacian of Gaussian filters) used in [36]. The largest filter window size is 19×19 pixels. We choose \mathbf{h} to be marginal histograms of filtered responses and sample the Julesz ensemble using Algorithm I. Although only a small subset of filters are often necessary for each texture pattern, we use a common filter set in this section. We shall discuss statistics

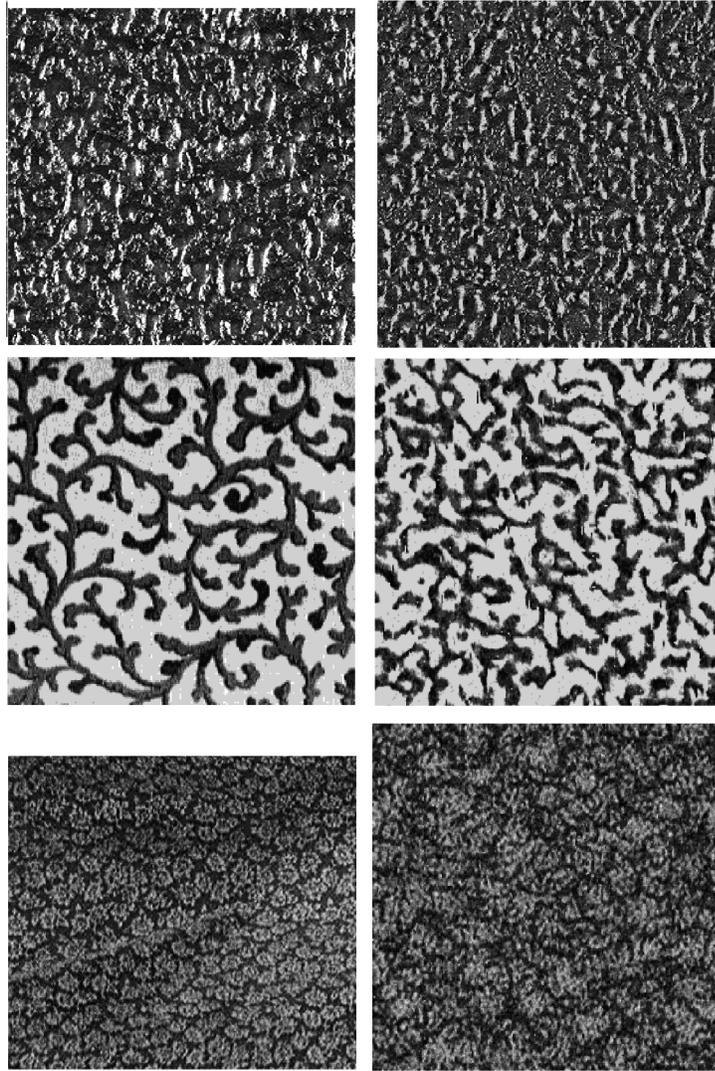


Fig. 6. Left column: the observed texture images. Right column: the synthesized texture images that share the exact histograms with the observed for 56 filters.

pursuit issues in Section 5. It is almost impractical to learn a FRAME model integrating all 56 filters in our previous work [36]; the computation is much easier using the simpler but equivalent model $q(\mathbf{I}; \mathbf{h})$.

We run the algorithm over a broad set of texture images collected from various sources. The results are displayed in Figs. 5, 6, 7, 8, and 9. The left columns show the observed textures and the right columns display synthesized images whose sizes are 256×256 pixels. For these textures, the marginal statistics closely match (less than 1 percent error for each histogram) after about 20 to 100 sweeps, starting with a temperature $T_0 = 3$. Since the synthesized images are finite, the matching error ϵ cannot be infinitely small. In general, we set $\epsilon \propto \frac{1}{|\Lambda|}$.

These experiments demonstrate that Gabor filters and marginal histograms are sufficient for capturing a wide variety of homogeneous texture patterns. For example, the cloth pattern in the middle row of Fig. 6 has very regular structures, which are reproduced fairly well in the synthesized texture image. Also, the crosses in Fig. 8 are synthesized without the special match filter used in [36].

This demonstrates that Gabor filters at various scales align up without using the joint histograms explicitly. The alignment or high order statistics are accounted for through the interactions of the filters.

Fig. 8 shows a periodic checkerboard pattern and a cheetah skin pattern. The synthesized checker board is not strictly periodic, and we believe that this is caused by the fast annealing process, i.e., the T in Algorithm I decreases too fast, so that the long range effect does not propagate across the image before local patterns form. The synthesized cheetah skin pattern is homogeneous whereas the observed pattern is not.

Our experiments reveal two problems:

1. The first problem is demonstrated in the two failed examples in Fig. 9. The observed texture patterns have large structures whose periods are longer than the biggest Gabor filter windows in our filter set. As a result, these periodic patterns are scrambled in the two synthesized images, while the basic texture features are well-preserved.

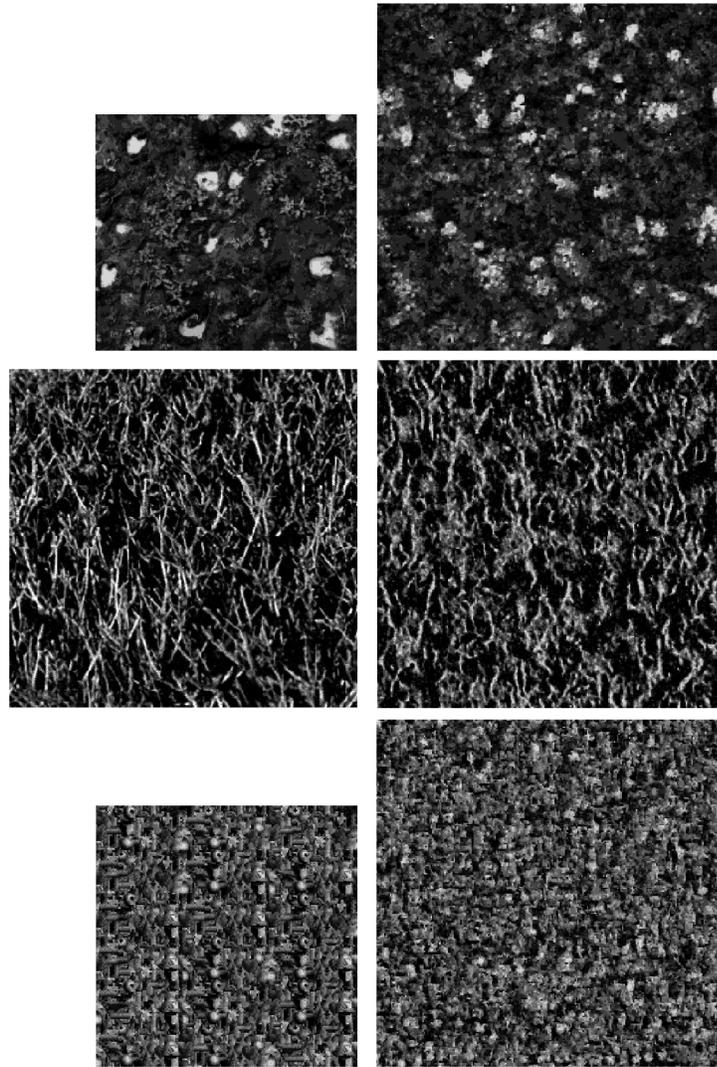


Fig. 7. Left column: the observed texture images. Right column: the synthesized texture images that share the exact histograms with the observed for 56 filters.

2. The second problem is with the effectiveness of the Gibbs sampler. If we scale up the checker board image so that each square of the check board is 15×15 pixels in size, then we have to choose filters with large window sizes. It becomes infeasible to match the marginal statistics closely using the Gibbs sampler in Algorithm I, since flipping one pixel at a time is inefficient for such large patterns. This suggests that we should search for more efficient sampling methods that can update large image patches. We believe that this problem would occur for other statistics matching methods, such as steepest descent [12], [2]. The inefficiency of the Gibbs sampler is also reflected in its slow mixing rate. After the first image is synthesized, it takes a long time for the algorithm to generate an image which is distinct from the first one. That is, the Markov chain moves very slowly in the Julesz ensemble. We shall discuss the effectiveness of sampling in Section 6.2.

5 SEARCHING FOR SUFFICIENT AND EFFICIENT STATISTICS

In this section we compare various sets of statistics discussed in Section 2, for their sufficiency in characterizing textures.

First, the full joint histogram defined in (1) appears to be an over-fit for most of the natural texture patterns. Thus, an MCMC algorithm matching the joint statistics generates texture images from a *subset* of the *true* Julesz ensemble. Our argument is based on two observations. 1) Our experiments partially shown in Section 4 demonstrate the sufficiency of marginal statistics. 2) Suppose that one uses a modest number of filters, e.g., 10 filters, and suppose that each filter response is quantized into 10 bins, then the joint histogram has 10^{10} bins. But one often has only a 128×128 texture image as training data. There are far too few pixels to estimate the full joint statistics reliably.

Our conclusion about the sufficiency of marginal statistics should not be overstated. We believe that this conclusion is only valid for general texture appearance. For example, one may construct a texture pattern with hundreds of

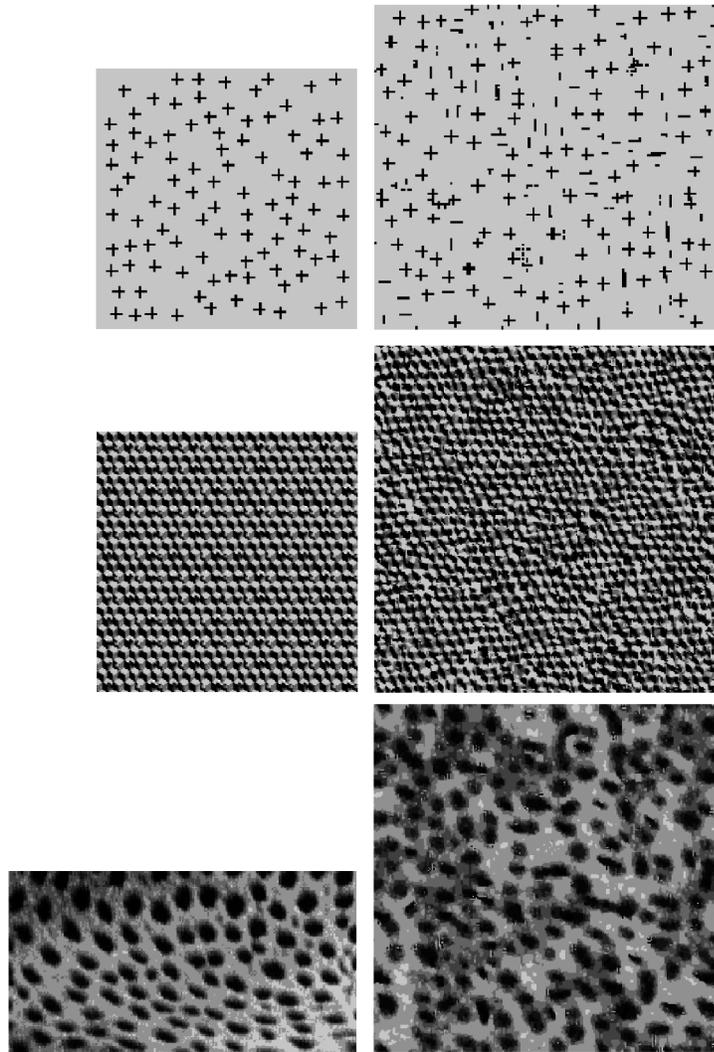


Fig. 8. Left column: the observed texture images. Right column: the synthesized texture images that share the exact histograms with the observed for 56 filters.

human faces as texture elements, where the detailed alignment of eyes, mouths, and noses are semantically important. Image features extracted by deformable face templates will become prominent. Indeed, the statistics extracted by templates can be considered as the general statistics extracted by some polygons across the image pyramid in (2).

In general, it is still possible that joint statistics crossing a small number of (say, $2 \sim 3$) filters at some bins (see Fig. 2d), i.e., not the full joint histogram or full co-occurrence matrices, are important for some image features, especially for texon patterns with deliberate details. Unfortunately, the number of combinations of such statistics grows exponentially in the statistics pursuit procedure. So we leave this topic for future research.

Second, we have computed the mean and variance for each of the subband images $\mathbf{h}^{(\alpha,i)}$, $\alpha = 1, 2 \dots, K, i = 1, 2$. As one may expect, the image matching the $2K$ statistics have noticeable differences from the observed ones. To save space, we shall not show the synthesized images.

Third, we have also calculated the two rectified functions $\mathbf{h}^{(\alpha,+)}, \mathbf{h}^{(\alpha,-)}$, $\alpha = 1, 2 \dots, K$. Although the rectified functions emphasize the tails of the histograms, which often correspond to texture features with large filter responses, the random images from the Julesz ensemble have noticeable differences from the observed.

Fig. 10 displays an experiment of texture synthesis using the rectified functions. The observed texture is shown in Fig. 11a and we match $\mathbf{h}^{(\alpha,+)}, \mathbf{h}^{(\alpha,-)}$ for all 56 filters used before. We varied the parameter θ in R^-, R^+ , so that $100 * \theta$ percent of the pixels lie between $[R^-, R^+]$. We display random examples from the Julesz ensembles with $\theta = 0.2, 0.8$, respectively.

Fourth, we proceed to study statistics which are simpler than marginal histograms. In particular, we are interested in knowing how many histogram bins are necessary for texture synthesis and how the bins are distributed. We adopt the filter pursuit method developed by Zhu et al. [36]. There are 56 filters in total and each histogram has 11 bins except for the intensity filter that has 8 bins. The algorithm sequentially selects one bin

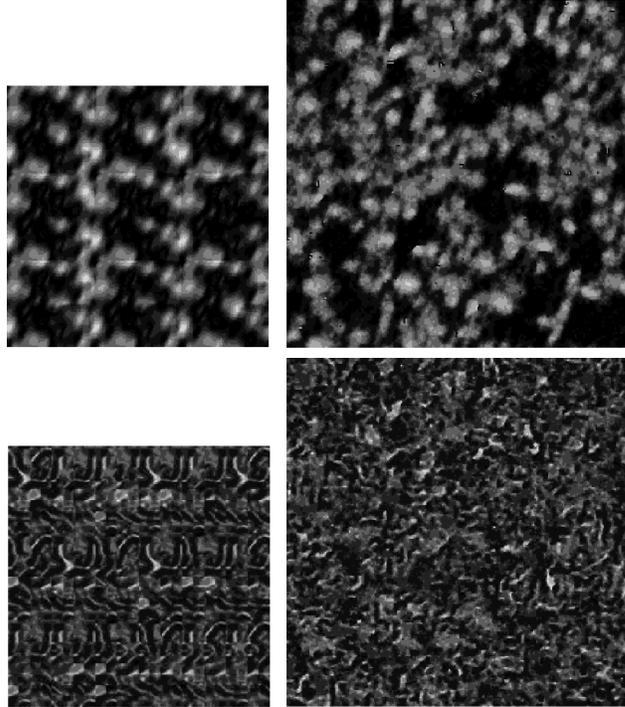


Fig. 9. Left column: the observed texture images. Right column: the synthesized texture images that share the exact histograms with the observed for 56 filters. The large periodic patterns are missing in the synthesized images due to the lack of large filters.

at a time according to an information criterion.⁶ Suppose m bins have been selected, and \mathbf{I} is a sample of the Julesz ensemble matching the m bins, then the information in the i th bin of the histogram $\mathbf{h}^\alpha(b; \mathbf{I})$ is measured by the matching error of this bin in a quadratic form,

$$d(\mathbf{h}^\alpha(b)) = \frac{(\mathbf{h}^\alpha(b; \mathbf{I}) - \mathbf{h}_{\text{obs}}^\alpha(b))^2}{2\sigma_\perp^2}. \quad (9)$$

In the above equation, σ_\perp^2 is the variance of $\mathbf{h}^\alpha(b)$ decorrelated with the previously selected statistics (see the appendix of [36]). Intuitively, the bigger the difference is, the more information the bin carries about the texture pattern. $d(\mathbf{h}^\alpha(b))$ is a second order Taylor approximation to the entropy decrease $\Delta_{\text{entropy}}(\mathbf{h}^\alpha(b))$ (see (5)) in the FRAME models. Because the entropy rate of the Julesz ensemble is the same as that of the FRAME model, $d(\mathbf{h}^\alpha(b))$ also measures the entropy decrease in the Julesz ensemble. Details about the approximation and the computation of σ_\perp^2 can be found in [36]. Since $d(\mathbf{h}^\alpha(b))$ is always positive, this means that as more statistics are used, the model becomes more accurate. This is not true when we only have finite observations, because we cannot estimate the observed histograms and variances exactly. Thus, the information gain $d(\mathbf{h}^\alpha(b))$ is balanced by a model complexity term that accounts for statistical fluctuations, such as the AIC criterion discussed in [36].

Because the second order approximation is only good locally, in the first few pursuit steps, we use the L_1 distance $d(\mathbf{h}^\alpha(b)) = \|\mathbf{h}^\alpha(b) - \mathbf{h}_{\text{obs}}^\alpha(b)\|_1$, and then use the quadratic distance after the sum of the matching errors for all the bins is below a certain threshold in terms of L_1 distance.

6. There is one redundant bin for each histogram.

Fig. 11 displays one example of bin selection. The left image is the observed image and it was originally synthesized by matching the marginal histograms of eight filters to a real texture pattern. We use this synthesized image as the observation, since it provides a ground truth in statistics selection. The right image is a sample from the Julesz ensemble using 34 bins. The selected bins for six filters are shown in Fig. 12. The height of each bin reflects the information gains; the higher bins are more important. The other two filters are the intensity filter (all eight bins are chosen) and the ∇_x filter whose bin selection is very similar to ∇_y .

Our bin pursuit experiments reveal two interesting facts: First, only a subset of bins are necessary for many textures. Second, the selected bins are roughly divided into two categories. One includes the bins near the center of histograms of some small filters, such as the gradient filters and Laplacian of Gaussian filters. These central bins enforce smooth appearances. The other bins are near the two tails of some large filters, which create image patterns. Such observations seems to confirm the design of Gibbs reaction-diffusion equations [37], where the central bins stand for diffusion effects and the tail bins for reaction effects.

6 DESIGNING EFFICIENT MCMC SAMPLERS

In this section, we study methods for efficient Markov chain Monte Carlo sampling.

6.1 A Generalized Gibbs Sampler

Improving the efficiency of MCMC has been an important theme in statistics and many strategies have been proposed in the literature [16], [17]. However, there are no good

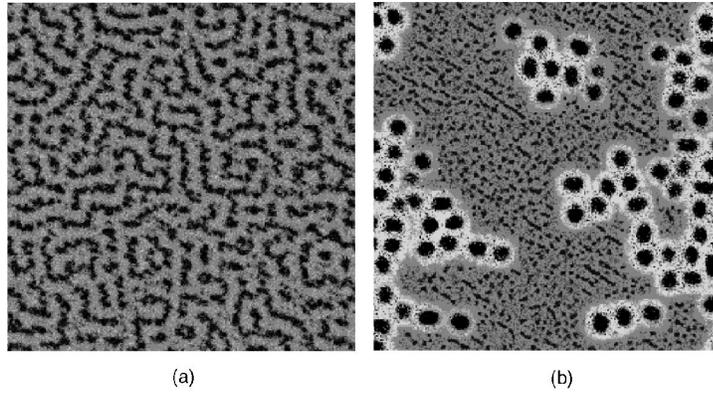


Fig. 10. Two sampled texture pattern from the Julesz ensemble matching the rectified functions of 56 filters. (a) $\theta = 0.2$ and (b) $\theta = 0.8$.

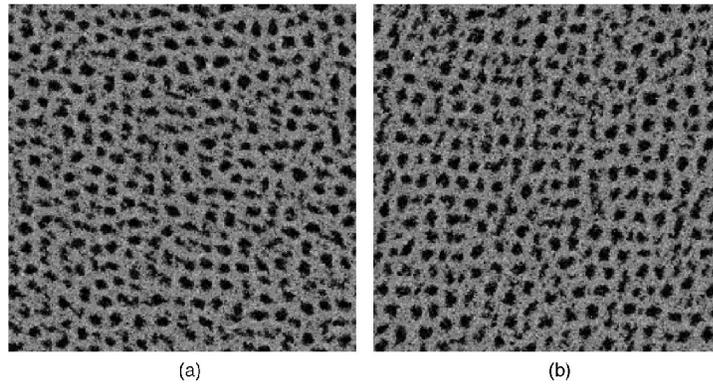


Fig. 11. An example of texture synthesis by a number of bins. (a) The observed image. (b) The sampled image using 34 bins.

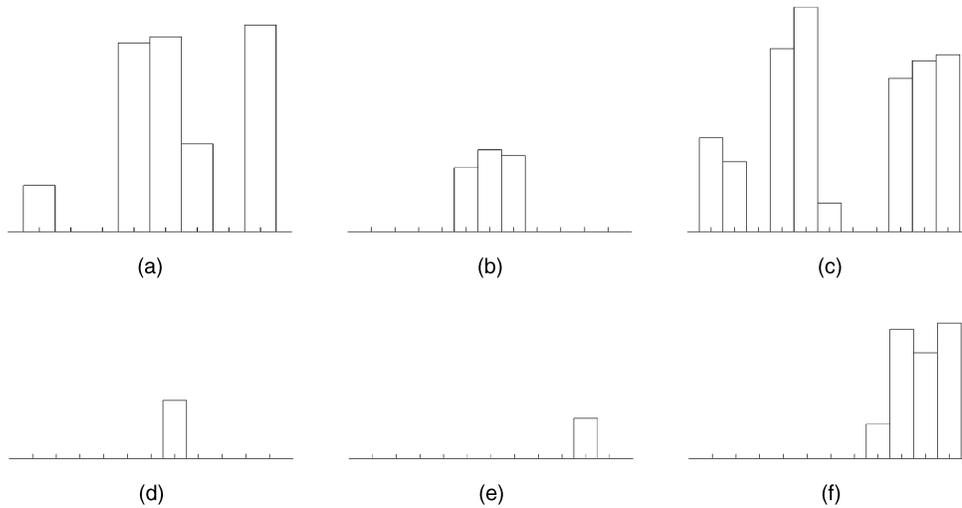


Fig. 12. The selected bins for six filter histograms, and the height of each bin reflects the information gain. (a) ∇_y , (b) LG(2) 9×9 pixels, (c) LG(4) 17×17 pixels, (d) $G_{cos}(4, 30^\circ)$, 11×11 pixels, (e) $G_{cos}(4, 90^\circ)$, 11×11 pixels, and (f) $G_{cos}(4, 150^\circ)$, 11×11 pixels.

strategies that are universally applicable, nor is there currently a satisfactory theory that can guide the search for better MCMC algorithms. In fact, the design of good MCMC moves often comes from heuristics of the problem domain. In this section, we propose a *window Gibbs* algorithm for sampling image models. Our algorithm is motivated by the recent work of Liu and Wu [24] and Liu and Sabatti [25].

Let $\mathbf{I}(\cdot, t) \in \Omega_\Lambda$ be a 2D image visited by the Markov chain at time t . For any pixel $v \in \Lambda$, $\mathbf{I}(v, t)$ denotes the

intensity values of the pixel at time t . We observe that the Gibbs sampler in Algorithm I adopts the following moves in the image space. At each step t , a pixel $v = (x, y) \in \Lambda$ is visited with a uniform distribution. Then, it moves the intensity value at v by Δ , with Δ sampled from a conditional probability.

$$\mathbf{I}(\cdot, t) \rightarrow \mathbf{I}(\cdot, t + 1) = \mathbf{I}(\cdot, t) + \Delta \cdot \delta_v.$$

$\delta_v \in \Omega_\Lambda$ is an image matrix on Λ which has intensity one at pixel v and zero everywhere else. It is a Dirac delta function

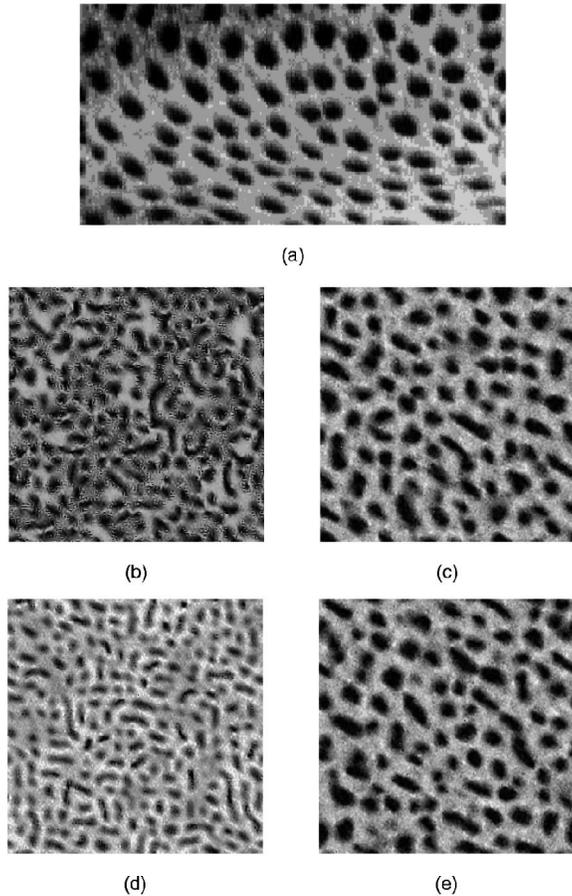


Fig. 13. (a) is the observed cheetah skin texture pattern, the image size of this image is as four times large as the one in Fig. 8. (b) and (d) are two sampled images using single site Gibbs sampler starting with a uniform noise image (b) and a constant image (d), respectively. (c) and (e) are the sampled images using the window Gibbs sampler starting with the same images as in (b) and (d), respectively.

translated to v and represented by a matrix. This algorithm is often called *single-site Gibbs*, as it moves along the dimensions of single pixel intensity.

As δ_v is an axis in Ω_Λ , it is natural to generalize single-site Gibbs by moving along other axes by the amount of $\Delta^{(\alpha)}$,

$$\mathbf{I}(\cdot, t) \rightarrow \mathbf{I}(\cdot, t+1) = \mathbf{I}(\cdot, t) + \Delta^{(\alpha)} W_v^{(\alpha)}, \quad (10)$$

In (10), $W_v^{(\alpha)}$ is a window function centered at pixel $v \in \Lambda$, and for convenience we choose $W^{(\alpha)}$ to be the window of filter $F^{(\alpha)}$ used for extracting texture features. $W_v^{(\alpha)}$ is a 2D matrix by translating the center of the window $W^{(\alpha)}$ to v and setting pixels outside the window to zero.

In the new proposed moves, as long as the intensity filter $\delta(\cdot)$ is included as one of the window functions, then one of the basic move is to flip the intensity at a single pixel. Thus, the Markov chain will be ergodic and aperiodic and has a unique invariant distribution on Ω_Λ . We call such a generalized Gibbs sampler the window Gibbs sampler because it updates pixels within a window along the direction of the window function.

In the window Gibbs sampler, one sweep visits each pixel $v \in \Lambda$ once, and at each pixel, a window $W^{(\alpha)}$ is

selected from the set of K filters according to a probability $p(\alpha)$. For example, we compute

$$p_1(\alpha) = \frac{\|\mathbf{h}^{(\alpha)}(\mathbf{I}) - \mathbf{h}_{\text{obs}}^{(\alpha)}\|_1}{\sum_{\beta=1}^K \|\mathbf{h}^{(\beta)}(\mathbf{I}) - \mathbf{h}_{\text{obs}}^{(\beta)}\|_1}, \quad \alpha = 1, 2, \dots, K. \quad (11)$$

$\|\cdot\|_1$ denotes L_1 distance. The filters with large matching errors have more chance to guide the direction of the move. Another choice for $p(\alpha)$ can be

$$p_2(\alpha) = \frac{\|\log \mathbf{h}^{(\alpha)}(\mathbf{I}) - \log \mathbf{h}_{\text{obs}}^{(\alpha)}\|_1}{\sum_{\beta=1}^K \|\log \mathbf{h}^{(\beta)}(\mathbf{I}) - \log \mathbf{h}_{\text{obs}}^{(\beta)}\|_1}, \quad (12)$$

which emphasizes matching the tails of the histogram $\mathbf{h}^{(\alpha)}(\mathbf{I})$.

Once α is selected, the value of $\Delta^{(\alpha)}$ in (10) is sampled from the conditional distribution

$$\Delta^{(\alpha)} \sim p(\Delta^{(\alpha)}) = \frac{q(\mathbf{I}(\cdot, t) + \Delta^{(\alpha)} W_v^{(\alpha)}; \mathbf{h})}{\sum_{\Delta} q(\mathbf{I}(\cdot, t) + \Delta W_v^{(\alpha)}; \mathbf{h})}.$$

In the denominator, the summation is over the centers of bins in the marginal histogram $\mathbf{h}_{\text{obs}}^{(\alpha)}$, i.e., we quantize Δ .

It is interesting to see that the computational complexity for computing $p(\Delta^{(\alpha)})$ is almost the same as computing $p(\mathbf{I}(v) | \mathbf{I}_{-v})$ in Algorithm I. In the Appendix, we briefly discuss the implementation details for computing the conditional distributions $p(\Delta^{(\alpha)})$.

A move in the window Gibbs algorithm essentially updates the projection of \mathbf{I} along $W_v^{(\alpha)}$, while fixing the projections of \mathbf{I} along all directions that are orthogonal to $W_v^{(\alpha)}$. So it is a conditional move or an ordinary Gibbs move seen from an orthogonal basis with one base vector being $W_v^{(\alpha)}$.

6.2 Experiment on the Window Gibbs Sampler

This section compares the performance of the window Gibbs sampler against the single-site Gibbs sampler.

Fig. 13a displays a cheetah skin texture. This image has been used in Fig. 8, but the image size in this experiment is four times as large as the one used before. The marginal histograms of eight filters are chosen as the statistics \mathbf{h} . Both the single-site Gibbs and the window Gibbs algorithms are simulated with two initial conditions: \mathbf{I}_u , a uniform noise image and \mathbf{I}_c , a constant white image. We monitor the total matching error at each sweep of the Markov chain,

$$E = \sum_{\alpha=1}^K \|\mathbf{h}^{(\alpha)}(\mathbf{I}_{\text{syn}}) - \mathbf{h}_{\text{obs}}^{(\alpha)}\|_1.$$

Fig. 13 displays the results for the first 100 sweeps. Fig. 13b and Fig. 13d are the results of the single site Gibbs starting from \mathbf{I}_u and \mathbf{I}_c , respectively. Fig. 13c and Fig. 13e are the results of the window Gibbs starting from \mathbf{I}_u and \mathbf{I}_c , respectively. The change of E is plotted in Fig. 14 against the number of sweeps. The dash-dotted curve is for the single site Gibbs starting from \mathbf{I}_u and the dotted curve is for the single site Gibbs starting from \mathbf{I}_c . In both cases, the matching errors remain very high. The two dashed curves are for the window Gibbs starting from \mathbf{I}_c and \mathbf{I}_u , respectively. For the latter two curves, the errors drop under 0.08. That means less than 1 percent error for each

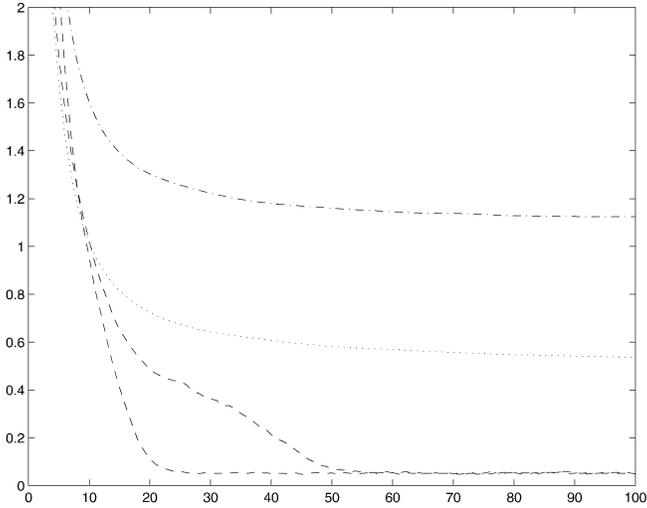


Fig. 14. The statistics matching error in L_1 distance summed over eight filters. The horizontal axis is the number of sweeps in MCMC (see text for explanation).

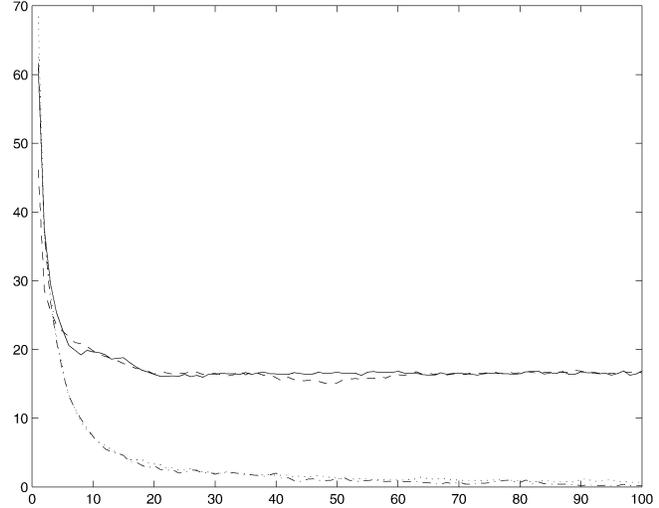


Fig. 15. The averaged intensity differences by MCMC in one sweep (see text for explanation).

histogram on average. The one starting from \mathbf{I}_u drops faster than the one from \mathbf{I}_c .

We now use another measure to test the effectiveness of the window Gibbs algorithm. We measure the Euclidean distance that the Markov chain travels during one sweep,

$$D(t) = \sqrt{\frac{1}{|\Lambda|} \sum_{v \in \Lambda} (\mathbf{I}(v, t) - \mathbf{I}(v, t - |\Lambda|))^2}.$$

Fig. 15 shows $D(t)$ for the single-site Gibbs (dash-dotted and dotted curves) and the window Gibbs (solid and dashed curves) starting from $\mathbf{I}_u, \mathbf{I}_c$, respectively.

In summary, the window Gibbs algorithm outperforms the single-site Gibbs in two aspects: 1) The window Gibbs can match statistics faster, particularly when statistics of large features are involved. 2) The window Gibbs moves faster after statistics matching. Thus, it can render texture images of different details.

A rigorous theory for why the window Gibbs is more effective has yet to be found. We only have some intuitive ideas. We believe that the window functions provide better directions than the coordinate directions employed by the single-site Gibbs for sampling $q(\mathbf{I}; \mathbf{h}, T)$ in two aspects. One is that it is easier to move towards a mode of $q(\mathbf{I}; \mathbf{h}, T)$ along these directions because they lead to substantial changes in

local filter responses, so the local spatial features can be formed very quickly. The other aspect is that it is easier to escape from a local mode along these directions, which is important especially in the low temperature situation. Of course, it is likely that these two aspects may require the use of different window functions.

6.3 Discussion of Other MCMC Methods

Equation (10) provides a way to design new families of Markov chains. We briefly discuss a few issues that are worth exploring in further research.

1. What is the optimal design for the directions of the Gibbs sampler? These directions may not have to be linear and can be any transformation groups [24].
2. In the window Gibbs algorithm, we adopted a simple heuristic, i.e., the probability $p(\alpha)$ for choosing the directions of moves nonuniformly. In general, one can choose other nonuniform probabilities, as well as statistics other than those used in the Julesz ensemble to drive the Markov chain. For example, one may select the joint statistics (histograms) as the proposal probability $q(\Delta^{(1)}, \Delta^{(2)}, \dots, \Delta^{(K)})$ to move the coefficients jointly. Such moves may be capable of creating texture elements (or textons) quickly at a location v because of the alignment of many filters.



Fig. 16. Texture patterns with flows.

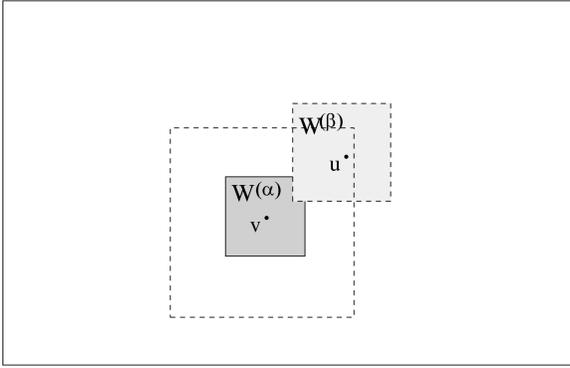


Fig. 17. The computation for flipping a window function at v . The response of filter $F^{(\beta)}$ at u needs to be updated if the two windows $W^{(\alpha)}$ at v and $W^{(\beta)}$ at u overlaps. The big dashed rectangle covers the pixels u whose filter responses should be updated.

7 CONCLUSION

In this paper, each Julesz ensemble on Z^2 is defined as a texture pattern, and texture modeling is posed as an inverse problem. The remaining difficulty in texture modeling is the search for efficient statistics that characterize texture patterns.

Although Section 2 provides a finite set of features and statistics, selecting the most efficient set of statistics within this dictionary is tedious. In particular, if we consider the general joint histogram bins in (10), we face an exponentially combinations. Furthermore, as the filter set is often over-complete, many different combinations of statistics can produce similar results.

Many texture patterns contain semantic structures presented in a hierarchical organization, as discussed in Marr's primal sketch [26]. For example, Fig. 16 shows three texture patterns: the hair of a woman, the twig of a tree, and zebra stripes. In these texture images, the basic texture elements and their spatial relationships are clearly perceivable. Indeed, the perception of these elements plays important role for precise texture segmentation. We believe that modeling textures by filters and histograms is only a first order approximation and further study has to be done to account for the hierarchical organization of the texture elements. This implies that we should look for meaningful features and statistics in a geometric hierarchy outside the current dictionary.

The study of image ensembles has significant applications beyond just texture modeling and synthesis. Object shapes [38] and other image patterns, such as clutter [37], can also be studied using the concept of ensembles. For example, Zhu and Mumford [37] studied two ensembles: tree clutter and images of buildings, and thus, they can separate the two patterns using Bayesian inference. Furthermore, the typical images in a given applications should be studied in order to design efficient algorithms and to analyze the performance of algorithms. Recently, Yuille and Coughlan [34] have applied the concept of ensembles to derive fundamental bounds on road detection.

APPENDIX

COMPUTING THE CONDITIONAL PROBABILITY

This appendix presents the implementation details for computing the conditional probability $p(\Delta^{(\alpha)})$ in the window Gibbs sampler.

Suppose that a pixel $v \in \Lambda$ and a filter α are selected at a certain step of the window Gibbs algorithm. The algorithm proposes a move

$$\mathbf{I}(\cdot, t) \longrightarrow \mathbf{I}(\cdot, t+1) = \mathbf{I}(\cdot, t) + \Delta^{(\alpha)} W_v^{(\alpha)}. \quad (13)$$

To compute the probability $p(\Delta^{(\alpha)})$, we need to update the histograms $\mathbf{h}^{(\beta)}$ for the subband image $\mathbf{I}^{(\beta)}(\cdot, t)$, $\beta = 1, 2, \dots, K$ with $\mathbf{I}(\cdot, t)$ being the state of the Markov chain at time t .

As shown in Fig. 17, the filter response $\mathbf{I}^{(\beta)}(u, t+1)$ at point $u \in \Lambda$ needs to be updated if the window $W_u^{(\beta)}$ overlaps with $W_v^{(\alpha)}$.

The new response of filter $F^{(\beta)}$ at u is

$$\mathbf{I}^{(\beta)}(u, t+1) = \mathbf{I}^{(\beta)}(u, t) + \Delta^{(\alpha)} \langle W_v^{(\alpha)}, W_u^{(\beta)} \rangle.$$

Therefore, we need to compute the inner products $\langle W_v^{(\alpha)}, W_u^{(\beta)} \rangle$ and save them in a table. Thus updating $\mathbf{I}^{(\beta)}(u)$ has only $O(1)$ complexity.

ACKNOWLEDGMENTS

We would like to thank the anonymous reviewers for the helpful comments that greatly improved the presentation of this paper. This work was supported by grant DAAD19-99-1-0012 from the U.S. Army Research Office and a U.S. National Science Foundation grant, IIS-98-77-127.

REFERENCES

- [1] H. Akaike, "On Entropy Maximization Principle," *Applications of Statistics*, P.R. Krishnaiah, ed., pp. 27-42, Amsterdam: North-Holland, 1977.
- [2] C.H. Anderson and W.D. Langer, "Statistical Models of Image Texture," *Unpublished preprint*, Washington University, St. Louis, Mo., 1996. (Send email to cha@shifter.wustl.edu to get a copy).
- [3] J. Besag, "Spatial Interaction and the Statistical Analysis of Lattice Systems," *J. Royal Statistical Soc., Series B*, vol. 36, pp. 192-236, 1973.
- [4] J.A. Bucklew, *Large Deviation Techniques in Decision, Simulation, and Estimation*. New York: John Wiley, 1990.
- [5] T. Caelli, B. Julesz, and E. Gilbert, "On Perceptual Analyzers Underlying Visual Texture Discrimination: Part II," *Biological Cybernetics*, vol. 29, no. 4, pp. 201-214, 1978.
- [6] R. Chellapa and A. Jain eds. *Markov Random Fields: Theory and Application*. Academic Press, 1993.
- [7] C. Chubb and M.S. Landy, "Orthogonal Distribution Analysis: A New Approach to the Study of Texture Perception," *Computer Models of Visual Processing*, Landy et al. eds., MIT Press, 1991.
- [8] G.R. Cross and A.K. Jain, "Markov Random Field Texture Models," *IEEE Trans. Pattern Analysis Machine Intelligence*, vol. 5, pp. 25-39, 1983.
- [9] J.S. De Bonet and P. Viola, "A Non-Parametric Multi-Scale Statistical Model for Natural Images," *Advances in Neural Information Processing*, vol. 10, 1997.
- [10] I. Daubechies, *Ten Lectures on Wavelets*. Philadelphia: Society for Industry and Applied Math, 1992.
- [11] J. Daugman, "Uncertainty Relation for Resolution in Space, Spatial Frequency, and Orientation Optimized by Two-Dimensional Visual Cortical Filters," *J. Optical Soc. Am.*, vol. 2, no. 7, 1985.
- [12] A. Galgalowicz and S.D. Ma, "Model Driven Synthesis of Natural Textures for 3D Scenes," *Computers and Graphics*, vol. 10, pp. 161-170, 1986.

- [13] S. Geman and D. Geman, “Stochastic Relaxation, Gibbs Distributions and the Bayesian Restoration of Images,” *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 9, no. 7, pp. 721-741, 1984.
- [14] S. Geman and C. Graffigne, “Markov Random Field Image Models and their Applications to Computer Vision,” *Proc. Int’l Congress of Math.*, 1986.
- [15] H.O. Georgii, *Gibbs Measures and Phase Transition*. New York: de Gruyter, 1988.
- [16] W.R. Gilks and R.O. Roberts, “Strategies for Improving MCMC,” *Markov Chain Monte Carlo in Practice*, W.R. Gilks et al. eds., chapter 6, Chapman and Hall, 1997.
- [17] W.R. Gilks, S. Richardson, and D.J. Spiegelhalter, *Markov Chain Monte Carlo in Practice*. Chapman and Hall, 1997.
- [18] D.J. Heeger and J.R. Bergen, “Pyramid-Based Texture Analysis/Synthesis,” *SIGGRAPH*, 1995.
- [19] B. Julesz, “Visual Pattern Discrimination,” *IRE Trans. Information Theory*, vol. 8, pp. 84-92, 1962.
- [20] B. Julesz, “Toward an Axiomatic Theory of Preattentive Vision,” pp. 585-612, *Dynamic Aspects of Neocortical Function*, G. Edelman et al. eds., New York: Wiley, 1984.
- [21] B. Julesz, *Dialogues on Perception*. MIT Press, 1995.
- [22] A. Karni and D. Sagi, “Where Practice Makes Perfect in Texture Discrimination—Evidence for Primary Visual Cortex Plasticity,” *Proc. Nat’l Academy Science, U.S.*, vol. 88, pp. 4,966-4,970, 1991.
- [23] S. Kullback and R.A. Leibler, “On Information and Sufficiency,” *Ann. Math. Statistics*, vol. 22, pp. 79-86, 1951.
- [24] J.S. Liu and Y.N. Wu, “Parameter Expansion for Data Augmentation,” *J. Am. Statistical Assoc.*, vol. 94, pp. 1,254-1,274, 1999.
- [25] J.S. Liu and C. Sabatti, “Generalized Gibbs Sampler and Multigrid Monte Carlo for Bayesian Computation,” *Biometrika*, to appear.
- [26] D. Marr, *Vision*, New York: W.H. Freeman and Company, 1982.
- [27] R.W. Picard, I.M. Elfadel, and A.P. Pentland, “Markov/Gibbs Texture Modeling: Aura Matrices and Temperature Effects,” *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, pp. 371-377, 1991.
- [28] K. Popat and R.W. Picard, “Cluster Based Probability Model and its Application to Image and Texture Processing,” *IEEE Trans. Information Processing*, vol. 6, no. 2, 1997.
- [29] J. Portilla and E.P. Simoncelli, “Texture Representation and Synthesis Using Correlation of Complex Wavelet Coefficient Magnitudes,” *Proc. IEEE Workshop Statistical and Computational Theories of Vision*, 1999.
- [30] A. Treisman, “Features and Objects in Visual Processing,” *Scientific Am.*, Nov. 1986.
- [31] R. Vistnes, “Texture Models and Image Measures for Texture Discrimination,” *Int’l J. Computer Vision*, vol. 3, pp. 313-336, 1989.
- [32] J. Weszka, C.R. Dyer, and A. Rosenfeld, “A Comparative Study of Texture Measures for Terrain Classification,” *IEEE Trans. System Man, and Cybernetics*, vol. 6, Apr. 1976.
- [33] Y.N. Wu, S.C. Zhu, and X.W. Liu, “The Equivalence of Julesz and Gibbs Ensembles,” *Proc. Int’l Conf. Computer Vision*, Sept. 1999.
- [34] A.L. Yuille and J.M. Coughlan, “Fundamental Limits of Bayesian Inference: Order Parameters and Phase Transitions for Road Tracking,” *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 22, no. 2, pp. 160-173, Feb. 2000.
- [35] S.C. Zhu, Y.N. Wu, and D. Mumford, “FRAME: Filters, Random Field and Maximum Entropy:—Towards a Unified Theory for Texture Modeling,” *Int’l J. Computer Vision*, vol. 27, no. 2, pp. 1-20, Mar./Apr. 1998.
- [36] S.C. Zhu, Y.N. Wu, and D. Mumford, “Minimax Entropy Principle and Its Application to Texture Modeling,” *Neural Computation*, vol. 9, no 8, Nov. 1997.
- [37] S.C. Zhu and D. Mumford, “Prior Learning and Gibbs Reaction-Diffusion,” *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 19, no. 11, Nov. 1997.
- [38] S. C. Zhu, “Embedding Gestalt Laws in Markov Random Fields,” *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 21, no. 11, Nov. 1999.



Song Chun Zhu received the BS degree in computer science from the University of Science and Technology of China in 1991. He received the MS and PhD degrees in computer science from Harvard University in 1994 and 1996, respectively. He was a research associate in the Division of Applied Math at Brown University in 1996 and a lecturer in the Computer Science Department at Stanford University in 1997. He joined the faculty of the Department of Computer and Information Sciences at The Ohio State University in 1998, where he is leading the OSU Vision And Learning (OVAL) group. His research is focused on computer vision, learning, statistical modeling, and stochastic computing. He has published more than 40 articles on object recognition, image segmentation, texture modeling, visual learning, perceptual organization, and performance analysis.



Xiuwen Liu received the BEng degree in computer science in 1989 from Tsinghua University, Beijing, China, the MS degree in geodetic science and surveying in 1995 and in computer and information science in 1996, and the PhD degree in computer and information science in 1999, all from The Ohio State University. Currently, he is a research scientist in the Department of Computer and Information Science at The Ohio State University. From 1989 to 1993, he was with the Department of Computer Science and Technology at Tsinghua University. His current research interests include image segmentation, statistical texture modeling, neural networks, computational perception, and pattern recognition with applications in automated feature extraction and map revision.



Ying Nian Wu studied at the University of Science and Technology of China (USTC) from 1988 to 1992 as an undergraduate student with a major in computer science and technology. He did his graduate study in the Department of Statistics at Harvard University and received the PhD degree in statistics in 1996. From 1997 to 1999, he was an assistant professor in the Department of Statistics at the University of Michigan. He is now an assistant professor in the Department of Statistics at the University of California, Los Angeles. He was a visiting researcher at Bell Labs in the summers of 1996 and 1997. His research interests include statistical computing and computer vision.